

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

*Кафедра автоматизованих систем обробки інформації та управління*

УДК: 004

«До захисту допущено»

**В.о. завідувач кафедри**

О.А.Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**Дипломний проект**  
**на здобуття ступеня бакалавра**

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: «Інформаційна підтримка створення ієрархічної  
структури 3D-об'єктів для 3D-сцен»

**Виконав:**

студент 4 курсу, групи ІС- 351

Юсеф Алі

(прізвище, ім'я, по батькові)

(підпис)

**Керівник**

ст. викл. Проскура С.Л.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з  
графічної  
документації**

ст. викл., к.т.н. Сперкач М.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Рецензент**

доц., к.т.н. Кисленко Ю.І.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент

(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) 6.050101

«Комп'ютерні науки» («Інформаційні управляючі системи та технології»)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

О.А.Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Юсефу Алі  
(прізвище, ім'я, по батькові)

**1. Тема проекту** «Інформаційна підтримка створення

ієрархічної структури 3D об'єктів для 3D-сцен»

керівник проекту Світлана Леонидовна Проскура, ст. викл.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “16” квітня 2019 р. №1260-с

**2. Термін подання студентом проекту** “01” червня 2019 року

**3. Вихідні дані до проекту**

Технічне завдання

**4. Зміст пояснювальної записки**

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

## 5. Перелік графічного матеріалу

1. *Схема структурна діяльності*

2. *Схема структурна варіантів використання*

3. *Схема бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Схема структурна послідовності*

6. *Схема структурна компонентів програмного забезпечення*

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «19» березня 2019 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>01.04.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.04.2019</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>01.05.2019</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>04.05.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>06.05.2019</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>09.05.2019</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>12.05.2019</i>	
8.	<i>Налагодження програми</i>	<i>20.05.2019</i>	
9.	<i>Виконання графічних документів</i>	<i>23.05.2019</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>25.05.2019</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>29.05.2019</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2019</i>	
13.	<i>Подання ДП рецензенту</i>	<i>05.06.2019</i>	

Студент

\_\_\_\_\_ А. Юсеф  
(підпис)

Керівник проекту

\_\_\_\_\_ С.Л.Проскура  
(підпис)

[illegible]



# **Пояснювальна записка до дипломного проекту**

на тему: «Інформаційна підтримка створення ієрархічної структури  
3D-об'єктів для 3D-сцен»

---

---

Київ – 2019 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з шести розділів, містить 16 рисунків, 10 таблиць, 1 додаток, 27 джерел.

У дипломному проекті реалізована тема «Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен», метою якої є спрощення інтеграції 3D-асетів в 3D-оточення (сцену) створене в Unreal Engine 4, завдяки категоризації та сортуванню великого об'єму 3д асетів, спрощенню процесу імпорту асетів в бібліотеку та спрощенню процесу експорту асетів в ігровий редактор Unreal Engine 4.

Розділ «Загальні положення» описує процес діяльності та можливі варіанти використання даної системи.

В розділі «Інформаційне забезпечення» описана структура вхідних та вихідних даних, наведена структура бази даних.

В розділі «Математична забезпечення» наведено детальний опис математичних алгоритмів, що були використані.

В розділі «Програмне та технічне забезпечення» описані основні програмні засоби, які були використані для розробки даної програми, наведені технічні вимоги до системи, на якій буде запускатись програма, описана програмна архітектура, яка була обрана для розробки.

В технологічному розділі наведено інструкцію користувача та набір тестів, проведених для оцінки якості роботи програми.

АСЕТ, ТЕКСТУРА, МОДЕЛЬ, UNREAL ENGINE 4, ІКОНКА, ТЕГ, ГУСТИНА ТЕКСЕЛЯ, МАТЕРІАЛ, ДЕКАЛЬ, АЛЬФА, МОДУЛЬ, HDR1, LUT, IES, ТОН, RGB, HSL, АЛЬБЕДО, ШОРСТКІСТЬ, НОРМАЛЬ, МАСКА, LOD.

					ДП ІС-5228.1260-с.ПЗ								
		Прізвище	Підпис	Дата									
Розроб.		Юсеф А.			Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен			Літ.		Лист		Листів	
Перевірів.		Проскура С.Л.								2		66	
								КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351					
Н. кон.		Телишева Т.О.											
Затв.		Проскура С.Л.											

## ABSTRACT

**Structure and extent of the project.** The explanatory note of this graduation project consists of six sections, contains 16 drawings, 10 tables, 1 supplement, and 27 sources.

For this thesis the theme that was implemented is «Informational Support for Creating a 3D-Object Hierarchical Structure for 3D Environments», the purpose of which is to ease the implementation of 3D Assets in 3D environments created in the Unreal Engine 4, by categorizing and sorting a large number of 3D assets, simplifying the process of importing assets into the library and simplifying the process of exporting assets to the Unreal Engine 4 editor.

The general provisions section describes the activities and use cases of the system.

In the informational provision section, the input and output data were defined, and the database structure was shown.

The mathematical provision section provides a detailed description of the mathematical algorithms that have been used.

The software provision section describes the main tools used to develop the program, the technical requirements for the system on which the program will be launched, and the software architecture selected for development.

The technical provision section consists of the user manual, and a number of tests conducted to evaluate the quality assurance of the program.

ASSET, TEXTURE, MODEL, UNREAL ENGINE 4, THUMBNAIL, TAG, TEXEL DENSITY, MATERIAL, DECAL, ALPHA, MODULE, HDRI, LUT, IES, HUE, RGB, HSL, ALBEDO, ROUGHNESS, NORMAL, MASK, LOD.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

## ЗМІСТ

ВСТУП.....	5
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....</b>	<b>8</b>
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	8
1.1.1 <i>Опис процесу діяльності.....</i>	<i>8</i>
1.1.2 <i>Опис функціональної моделі.....</i>	<i>12</i>
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ.....	14
1.3 ПОСТАНОВКА ЗАДАЧІ.....	16
1.3.1 <i>Призначення розробки.....</i>	<i>16</i>
1.3.2 <i>Цілі та задачі розробки.....</i>	<i>16</i>
Висновок до розділу.....	16
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>17</b>
2.1 ВХІДНІ ДАНІ.....	17
2.2 ВИХІДНІ ДАНІ.....	18
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	18
Висновок до розділу.....	22
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>23</b>
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	23
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ.....	25
3.2.1 <i>Задача генерації середнього кольору із альbedo-текстури.....</i>	<i>25</i>
3.2.2 <i>Задача генерації тону із RGB-кольору.....</i>	<i>25</i>
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ.....	26
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ.....	26
3.4.1 <i>Генерація середнього кольору із альbedo-текстури.....</i>	<i>26</i>
3.4.2 <i>Генерація тону із RGB кольору.....</i>	<i>27</i>
Висновок до розділу.....	28
<b>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>29</b>
4.1 ЗАСОБИ РОЗРОБКИ.....	29
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ.....	30
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	31
4.3.1 <i>Діаграма класів.....</i>	<i>31</i>

4.3.2	Діаграма послідовності .....	32
4.3.3	Діаграма компонентів .....	34
4.3.4	Специфікація функцій.....	35
	Висновок до розділу .....	36
<b>5</b>	<b>ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>37</b>
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	37
5.1.1	Пошук .....	38
5.1.2	Імпорт .....	42
5.1.3	Експорт.....	47
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	48
5.2.1	Мета випробувань .....	48
5.2.2	Загальні положення.....	48
5.2.3	Результати випробувань .....	49
	Висновок до розділу .....	51
	ЗАГАЛЬНІ ВИСНОВКИ .....	52
	ПЕРЕЛІК ПОСИЛАНЬ .....	53
	ДОДАТОК А .....	56

## ВСТУП

Протягом останніх 20 років індустрія відеоігор активно набирає популярність. Статистичні дані представлені Reuters [1] ставлять ігрову індустрію на перше місце по прибутковості та популярності в розважальній сфері. За цими даними, в 2018 році прибуток від реалізації відео-ігр у світі склав 116 мільярдів доларів. За цей же рік сумарний прибуток отриманий телевізійними компаніями складає 105 мільярдів доларів. Типові представники шаленого успіху в індустрії можна знайти щороку. Так, наприклад, Grand Theft Auto V [2], випущена в 2013 році, принесла близько 6 мільярдів доларів прибутку компанії Rockstar Games [3] в період 2013-2018 років. При цьому кількість активних користувачів гри досі є дуже високою, отже вона продовжує приносити прибуток. На даний момент це один із найбільш успішних тайтлів в медіа-сфері.

З постійним ростом індустрії, ростуть і розвиваються малі студії розробки відео-ігр і великі компанії, що призводить до росту бюджетів, які витрачаються на розробку. Причому більша частина бюджету йде саме на заробітні плати спеціалістів (займає від третини до половини всього бюджету). Це означає, що час розробки - є найважливішим аспектом в процесі розробки. Час розробки ігрових проектів може тривати від декількох місяців до декількох років.

Для оптимізації періоду розробки в ігровій індустрії багато уваги приділяють оптимізації процесу роботи розробників. Використовуються такі засоби як:

- автоматизації повсякденних задач що часто повторюються;
- процедурна генерація середовища;
- аутсорсинг завдань.

Це далеко не всі методи які дозволяють зменшити час розробки без шкоди для якості фінального продукту.

					ДП ІС-5228.1260-с.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

При цьому автоматизації підлягає велика частина процесу розробки – від створення скриптів, що виконують об’ємні задачі без участі розробника (наприклад генерація місцевості), до створення програмного забезпечення, яке може пришвидшити деякі конкретні етапи розробки. Даний програмний продукт створений для того, щоб оптимізувати роботи будь-якого розробника на Unreal Engine 4 [4], починаючи від левел-дизайнера та 3D-художника оточення, який формує ігрове оточення на сцені, закінчуючи 3D-моделером, який створює та структурує 3D-моделі, та навіть програмістом, якому потрібно швидко отримати доступ до певних асетів [5] для тестування ігрових механік або створення функціонального прототипу майбутньої гри.

Дане програмне забезпечення спрощує процес створення 3D-асетів, слугуючи автоматизованим "каналом зв'язку" між стадією створення асету та його фінальною репрезентацією в редакторі Unreal Engine 4. Також дане програмне забезпечення може слугувати великою структурованою бібліотекою 3D-асетів, які можна швидко знайти, експортувати в редактор та використовувати для розробки ігрового оточення. Також дана бібліотека підтримує додавання нових асетів, що може дозволити розробникам мати під рукою велику бібліотеку контенту, яка динамічно оновлюється та може бути використана на багатьох проектах.

**Практичне значення одержаних результатів.** Була створена динамічна бібліотека 3D-асетів для розробки ігрового середовища.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ**

Асет – будь-яка текстура, 3D-модель, матеріал або інший формат файлів, що використовуються при створенні середовища;

UE4 – Unreal Engine 4;

TD – Texel Density;

HDRi – High-dynamic-range imaging;

LUT – Lookup-Table;

IES – Illuminating Engineering Society;

RGB – Red-Green-Blue;

HSL – Hue-Saturation-Lightness;

LOD – Level of detail.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7



# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Опис предметного середовища

На даний момент автоматизації підлягає велика частина процесу роботи ігрових розробників. Це дозволяє звільнити працівників від монотонних та однотипних задач та направити їх сили на більш креативні процеси.

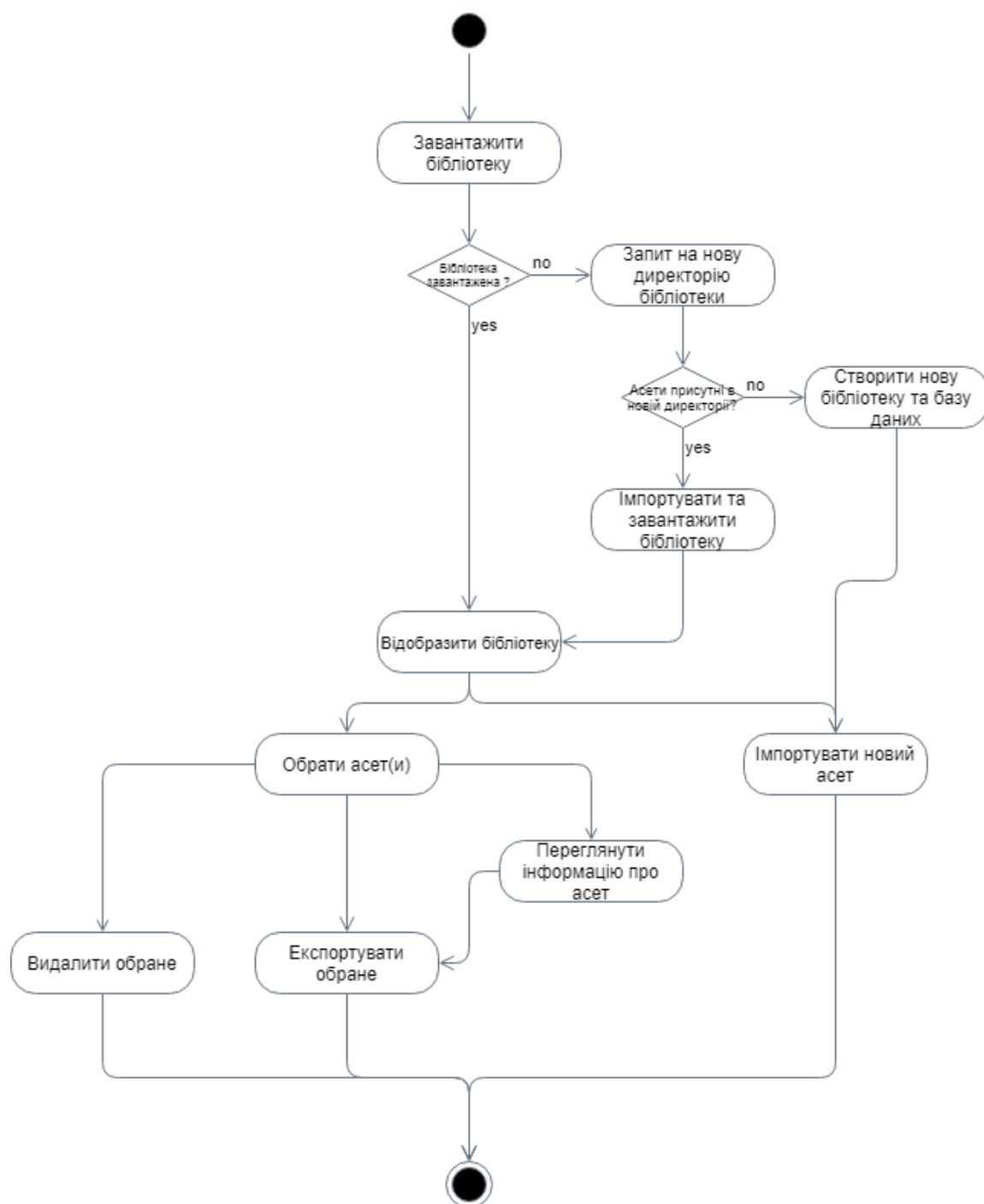
Одна з таких задач – це структуризація та конвертація асетів в певний формат, зі своєю системою іменування та ієрархією папок, що дозволяє використовувати дані асети в Unreal Engine 4 різними розробниками (лєвел-дизайнер, дизайнер оточення). При цьому структуризація є дуже важливою, адже в великих ігрових проектах сумарна кількість асетів, що використовується, може перевищувати десятки тисяч. Дуже важливо при імпорті в редактор зберегти асети таким чином, щоб в майбутньому не довелося корегувати їх структуру або іменування.

Даний проект концентрується на автоматизації процесу інтеграції щойно створених асетів в велику бібліотеку та на процесі їх експорту з бібліотеки в ігровий редактор в коректному вигляді для подальшого використання.

### 1.1.1 Опис процесу діяльності

На рисунку 1.1 зображена діаграма діяльності, яка ілюструє процес роботи бібліотеки 3D-асетів.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8



**Рисунок 1.1** – Схема структурна діяльності. Процес роботи бібліотеки 3D-асетів

Процес роботи починається зі спроби завантажити файли бібліотеки, якщо такі існують. Якщо бібліотека не існує (не вдається знайти асети) або є пошкодженою, програма пропонує користувачу ввести нову директорію для бібліотеки, а потім або завантажити існуючу бібліотеку в новий каталог, або створити базу даних і структуру бібліотеки всередині вибраного каталогу.

Після того, як бібліотека завантажена, користувач обирає одну з функцій – або імпорт нового асета, що додає його в бібліотеку, або вибір поточних асетів із бібліотеки та їх експорт до Unreal Engine 4.

Для процесу імпорту програма повинна отримати наступні дані від користувача:

- іконка, яка буде використовуватись для зображення асета в бібліотеці;
- прев'ю-зображення. Може бути в будь-якій кількості або бути відсутнім. Сюди вставляються скріншоти моделі з різних кутів, з або без детального освітлення, з текстурами та без і так далі. Для текстур, наприклад, правильним буде наводити скріншот текстури накладеної на площину;
- назва асету, яка буде відображатись в бібліотеці та в ігровому редакторі після експорту до нього;
- теги. Ключові слова, які спрощують процес пошуку та фільтрації асетів. Кожен асет може мати декілька тегів, які коротко (чим менше слів на тег, тим краще) описують даний асет. Наприклад камінь, трава, квітка, будинок і т.д.;
- коментарій (замітка) щодо проблемних або специфічних моментів, які пов'язані з даним асетом (наприклад проблема з текстурою, не коректні нормалі в моделі і т.д.). Опціонально;
- тип асету, який визначається методом його створення (фотореалістичний відсканований асет, процедурний асет або зроблений вручну;

- Texel Density [6] - параметр, оснований на розмірі текстури в пікселях (наприклад 2048x2048 пікселей) та розмірі текстури в одиницях виміру простору (наприклад текстура використовується для підлоги розміром 2x2 метри). Таким чином даний параметр залежить і від текстури, і від 3D-моделі, для якої текстура призначена. Густина текселів також залежить від покриття та перекриття UV-оболонок 3D-моделей, що означає, що цей процес не може бути автоматизованим. Інформація про щільність текселя потрібна лише тоді, коли об'єкт містить текстури;
- категорія – одна з декількох категорій, які використовуються для фільтрації асетів.

Другий метод використання даного програмного засобу полягає в пошуку та експорті асетів з бібліотеки. Асети можуть бути відсортовані, відфільтровані за типом, середнім кольором текстури, категорією, ключовими словами та мінімальним значенням густини текселів.

Під час пошуку асетів користувач може ознайомитись з інформацією про даний асет, експортувати один або декілька вибраних асетів в Unreal Engine 4. Користувач може змінювати налаштування експорту виходячи з вимог проекту. Дані налаштування включають:

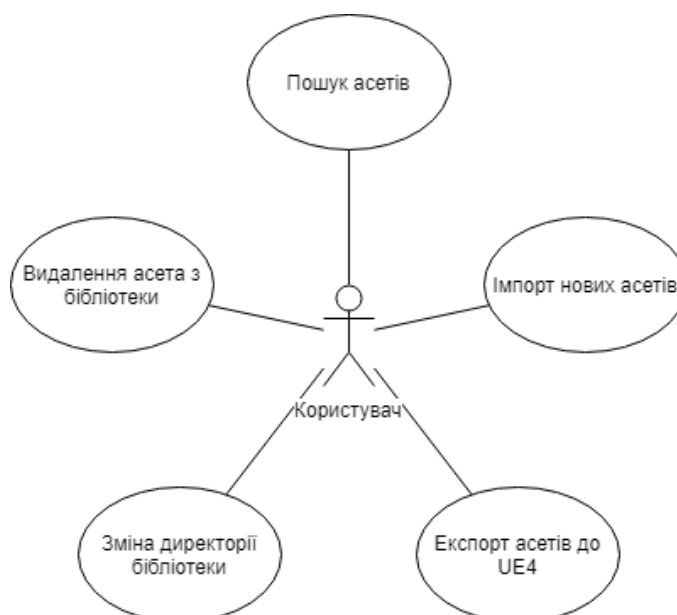
- розмір текстури в пікселях;
- пакування текстур – дозволяє упакувати декілька текстур в одне зображення для зменшення сумарного розміру;
- формат текстури: jpeg, png, targa і так далі;
- розмірність текстури: 8-bit, 16-bit і так далі. Обидва налаштування (формат та розмірність) використовуються для того щоб збалансувати розмір файлу текстури та налаштування компресії, які впливають на якість вихідної текстури;

- перевірка імен: зміна імен асетів основана на вимогах проекту, що дозволяє підтримувати чистоту в структурі файлів.

### 1.1.2 Опис функціональної моделі

Призначенням розробки є створення програмного забезпечення яке допоможе оптимізувати робочий процес 3D-художників, левел-дизайнерів та дизайнерів оточення. Більш того, в ролі всіх цих спеціалістів в невеликих студіях може виступати одна людина (розроблення 3д-асетів і, в той же час, створення ігрових рівнів та оточення). Саме тому програма спроектована таким чином, щоб її було зручно використовувати саме одній людині (тобто в програмі не реалізовано розділення прав між художником, який може лише імпортувати асети в програму, та дизайнером оточення, який може їх лише експортувати в редактор).

Відповідно визначених варіантів використання представлено загальну модель варіантів використання на рисунку 1.2.



**Рисунок 1.2** – Схема структурна варіантів використання. Варіанти використання системи користувачем

Типи залежностей між варіантами використання описані в таблиці 1.1.

**Таблиця 1.1**– Типи залежностей між варіантами використання

<i>Актор</i>	<i>Варіант використання</i>	<i>Опис дії варіанта використання</i>
Користувач	Пошук асетів	Користувач шукає асети в бібліотеці використовуючи фільтрацію, сортування та категорії.
	Імпорт нового асету	Користувач може додати новий асет до бібліотеки, якщо вся необхідна інформація та файли будуть доступні.
	Експорт асетів до Unreal Engine 4	Користувач може змінювати налаштування експорту, основуючись на яких проводиться автоматичний експорт асетів в проект
	Зміна директорії бібліотеки	Користувач може змінити директорію проекту, що також дозволяє створити пусту бібліотеку або обрати існуючу в певній папці.
	Видалення асету з бібліотеки	Видалення асету відправляє всі його файли до «локальної» кошика бібліотеки, яка може бути очищена через програму, або всі асети з неї можуть бути відновлені. Перенесені в кошик асети архівуються щоб економити дисковий простір, а записи про них з база даних видаляються.

## 1.2 Огляд наявних аналогів

Ігрова індустрія сьогодні використовує багато методів автоматизації робочого процесу розробників. Більша частина уваги концентрується на автоматизації створення асетів через процедурну генерації або програмовані програмні засоби / тули.

Ще один спосіб збереження бюджету і часу студії – аутсорсинг створення асетів, що дозволяє купувати їх в онлайн-магазинах або замовляти у студій які спеціалізуються саме на створенні асетів а не розробці ігор.

На даний момент є певна кількість великих магазинів, які по суті є онлайн-бібліотеками, що зберігають та продають асети, або надають їх для безкоштовного скачування. Один з таких – Unreal Marketplace [7], який належить компанії Epic Games [8] – розробникам Unreal Engine 4. На даному ресурсі розміщені тисячі асетів, паків асетів (моделі, текстури, матеріали, музика, 2D-арт і так далі) і плагінів для Unreal Engine 4. Більшість з цих асетів створені користувачами (відповідно вони получають дохід від продажу).

Після покупки асетів на маркеті вони залишаються назавжди прив'язаними до акаунта розробника на Unreal Engine 4, що дозволить використовувати їх завжди. Прямий експорт асетів в редактор дозволяє зекономити час на їх створенні і структуризації для використання в редакторі.

Фаза інтеграція асетів в ігровий редактор ігнорується багатьма розробниками, проте її автоматизація може зберегти багато часу перед початком створення 3D-середовищ для гри.

Саме тому важливо надати можливість експортувати підготовлені та категоризовані асети із програми напряму в проект на Unreal Engine 4. Таким чином розробник може швидко додавати нові асети та не відволікатись від процесу створення оточення.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Ще один приклад таких маркетів – це cgtrader [9] та turbosquid [10], які мають велику бібліотеку 3D-моделей і постійно оновлюють свою базу.

Ці сайти ідеальні для пошуку специфічних конкретних моделей які потрібні для середовища. Велика кількість асетів на сайтах пояснюється великою кількістю фрілансерів, які заробляють на життя створенням і продаванням цих асетів.

В таблиці 1.2 наведено порівняльний аналіз між описаними аналогами та даним програмним продуктом.

**Таблиця 1.2**– Порівняльний аналіз даної роботи та наявних аналогів

Функція	Дана робота	Unreal Engine Marketplace	CGTrader / Turbosquid
Підготовлені до експорту асети	Так	Так	Ні
Специфічні асети	Так	Ні, лише групи асетів	Так
Кастомізовані налаштування експорту	Так	Ні	Ні
Кастомізоване форматування асетів для кожного проекту	Так	Ні	Ні
Фільтрування асетів на основуючись на різних параметрах	Так	Ні	Так



### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначенням системи є організація та динамічна підтримка бібліотеки асетів, яку можна використовувати на безлічі проектів без необхідності перероблювати асети.

#### 1.3.2 Цілі та задачі розробки

Метою системи є пришвидшити процес створення 3D-середовища завдяки автоматизації інтеграції 3D-асетів в середовище та категоризації всіх асетів в бібліотеці.

Для досягнення поставленої мети мають бути вирішені такі задачі:

- створення категоризованої бібліотеки х пошуковою системою, яка включає пошук, фільтрацію та сортування;
- автоматизація всіх процесів які можуть бути автоматизовані в процесі імпорту асетів до бібліотеки;
- автоматизація з можливістю зміни налаштувань в фазі експорту з бібліотеки до редактора, що дозволяє ігнорувати фазу прямо імпорту в редактор і витрачання часу на підготовку асетів до імпорту;
- створити колекцію асетів які можна використовувати в різних проектах.

#### Висновок до розділу

В даному розділі були описані мета та цілі розробки, створені базові вимоги, які потрібно виконати при розробці програми. Було описано значення даної програми для користувача, наведені найближчі аналоги та порівняльний аналіз їх з даним програмним продуктом.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідні дані представлені у таблиці 2.1.

**Таблиця 2.1** – вхідні дані.

Данні	Опис
Інформація про асет	Інформація про асет що імпортується, така як його назва, категорія, теги і т.д. Ця інформація вводиться в форму застосування після початку імпорту.
Зображення асету	Дані зображення – це іконка асету для пошукової системи та прев'ю, яке детально зображує асет з усіх боків зі світлом та без, з текстурами та без, при детальному та звичайному освітлені і т.д. Асет повинен мати рівно одну іконку та будь-яку кількість прев'ю зображень, які показують як асет буде виглядати в редакторі. Зображення додаються в бібліотеку при імпорті асетів.
Файли асету	Моделі, текстури та інші файли із яких складається асет. Файли додаються при імпорті нового асету та використовуються в процесі фільтрації асетів пошуковою системою.
Ключові слова для фільтрації	Ключові слова, які вводяться в пошукову строку, допомагають відфільтрувати необхідні асети та швидше їх знайти.

## 2.2 Вихідні дані

Вихідні дані представлені в таблиці 2.2.

**Таблиця 2.2** – вихідні дані

<i>Данні</i>	<i>Опис</i>
Інформація про асет	Відображається вікно з інформацією про асет, куди входить інформація введена користувачем при імпорті та критерії фільтрації які визначає сама програма при імпорті асету.
Файли асету	При експорті в редактор файли упаковуються, перейменовуються та інколи переналаштовуються перед тим як потрапити в редактор.

## 2.3 Опис структури бази даних

База даних складається з однієї таблиці до якої входить інформація про всі асети, що були імпортовані в бібліотеку.

Окрім запису в таблиці, кожен асет має свій .ini файл, який зберігає ту ж інформація що й у базі даних, але представлений локально для кожного асету. Даний файл використовується при імпорті великої кількості асетів. Це дозволяє змінювати директорію бібліотеки, копіювати асети між різними бібліотеками, ділитись асетами з іншими машинами, синхронізувати бібліотеку з асетами на диску, що також дозволяє уникнути великої кількості помилок, таких як неспівпадання записів у базі з реальними файлами на диску, яке може виникнути в наслідок пошкодження або втрати локальних файлів.

Таблиця бази даних має всю необхідну інформацію для відображення та експорту асетів без необхідності «витягати» кожен асет вручну зі своєї

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

директорії. Реалізовані SQL-команди для пошуку, фільтрування і сортування асетів без необхідності читати та перевіряти інформацію по кожному файлу, що може займати дуже багато часу, коли бібліотека стане великою.

База даних читається динамічно. Це означає, що з бази в оперативну пам'ять зчитується інформація лише про ті асети, які зараз наявні в вікні пошуку. Це дозволяє зберегти пам'ять за рахунок меншої кількості іконок та прев'ю-зображень що підвантажуються.

Таблиця включає наступні колонки:

- ID: генерується автоматично, вказує на унікальний порядковий номер асету незважаючи на параметри фільтрації та сортування;
- AssetName: назва асета, використовується для виведення пошуковою системою та як ім'я нового асета під час експорту до редактора;
- AssetID: Використовується для категоризації всіх асетів та зберігання їх в структурованій директорії без необхідності читати інформацію про асет щоб отримати доступ до нього;
- Type: тип асету який може бути відсканований (використовується фотограмметрія, фотограмметричне стерео або інші технології фото реалістичного сканування), процедурний (коли мається процедурна текстура, модель, браш, процедурно створене в певному програмному забезпеченні дерево, або будь-який інший тип процедурності, коли асет може мати оригінальний файл, який можна використати для отримання нових варіацій даного асету) або зроблений вручну з використанням , фотограмметричне стерео або інші технології фото реалістичного сканування), процедурний (коли мається процедурна текстура, модель, браш, процедурно створене в певному програмному забезпеченні дерево, або будь-який інший тип процедурності, коли асет може мати оригінальний файл, який можна використати для отримання нових варіацій даного асету) або

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

зроблений вручну з використанням 3D-програмного забезпечення (не може бути рандомізованим та не є фото реалістичним);

- TD: вище описаний параметр густини текселів. Задається вручну, не може бути вирахований автоматично, так як залежить від реальних розмірів та параметрів моделі які можуть змінюватись;
- Colour: середній колір текстури асету, якщо текстура наявна. Розраховується автоматично під час імпорту асету до бібліотеки за рахунок пошуку середнього кольору всіх пікселів на першій текстурі типу albedo/diffuse якщо така присутня;
- Hue: вирахований колір асету конвертований із RGB до HSL. Даний параметр може бути використаний для фільтрування та сортування асетів основуючись на його значенні. Причина використання даного параметру для фільтрації а не всього кольору – можливість уникнути потрапляння параметрів насичення та яскравості в процес фільтрації;
- Tags: ключові слова що використовуються в процесі фільтрації та пошуку. Вводяться в ручну на стадії імпорту;
- Notes: Необов'язкове поле, куди записується інформації про проблемні або специфічні моменти пов'язані з асетом;
- Resolution: розмірність текстури асету в пікселях, якщо текстура наявна. Вираховується автоматично як розмірність текстури мінімальної розмірності, яка була імпортована з асетом;
- Category: асет може мати одну з таких категорій:
  - 1) Materials: колекція текстур, які можуть покривати безкінечну площину постійно повторюючись, уникаючи «швів» на стиках текстур;
  - 2) Decals: колекція текстур з альфа каналом, які можуть бути розміщені поверх інших об'єктів в середовищі (текстура калюжі,

- пятно на одязі, вибита зона на цегляній стіні будинку – це все decals);
- 3) Alphas: чорно-біле зображення яке може бути використане як маска в матеріалах;
  - 4) Trees: процедурні 3D-дерева, які підтримують симуляцію вітру;
  - 5) Foliage: різні рослини, квіти, трава та інші моделі рослинності, якими заповнюється оточення;
  - 6) Organic: моделі природи, такі як каміння, мертві дерева, пеньки і так далі;
  - 7) Man-Made: об'єкти на твердій поверхні, транспорт, фурнітура, зброя – в цілому всі об'єкти створені людиною а не природою;
  - 8) Architecture: модульні стіни, будинки і навіть замки – все що відноситься до створення будівельних структур або є такою структурою;
  - 9) Terrain: гори, долини, рівнини і все що представляє собою великий пустий ландшафт який слугує як основа майбутнього 3D-оточення або як фон;
  - 10) Modules: невеликі частини моделей, які можна використовувати знову і знову для скорочення часу моделювання певних асетів (наприклад при моделюванні автомобілей, незважаючи на модель авто, можна зробити лише одну модель шин, а потім використовувати її знову і знову);
  - 11) HDRi: (*High-Dynamic-Range imaging*) панорамне зображення з HDR можливостями яке використовується для освітлення середовища (наприклад небо); [11]
  - 12) LUT: (*Look Up Tables*) текстура, яка використовується як база для класифікації кольорів. Використовується для пост-обробки візуальної частини середовища; [12]

- 13) IES: (*Illuminating Engineering Society*) – дана структура використовується для стандартизації того, як джерело світла моделює освітлення, що є симуляцією реальних типів освітлення в житті; [13]

На рисунку 2.1 представлена структурна схема бази даних.

Assets	
id 🔗	int
AssetName	string
AssetID	string
Type	string
TD	int
Colour	Colour
Category	string
Tags	string
Notes	string
Resolution	string
Hue	float

**Рисунок 2.1** – Схема бази даних

### Висновок до розділу

У даному розділі було розглянуто всі вхідні та вихідні дані, була описана структура таблиці бази даних.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Один з параметрів для фільтрування та сортування – це головний колір асету, генерація якого може бути автоматизована щоб зберегти час користувача від ручного введення кольору під час імпорту асетів.

Альbedo-текстура це кольорова інформація про асет, яка не включає інформацію про тіні або відображення, що робить її ідеальною базою для обчислення середнього кольору асету.

Обраний метод сортування та фільтрації використовує головну альbedo-текстуру асету (якщо вона існує) щоб згенерувати середній колір, який потім конвертується із RGB до HSL та зберігається в базу даних.

RGB створюється із таких каналів:

- червоний;
- зелений;
- синій;
- альфа: опціональний канал, який показує прозорість пікселя.

Кожний піксель має число, яке репрезентує як ці три кольори та альфа-канал змішуються, створюючи необхідний колір.

HSL конструється із наступних каналів:

- hue (тон): число, яке представляє домінуючу довжину хвилі кольору, також може бути представлене як координата кольорового спектра або колеса кольору;
- saturation: інтенсивність кольору, вираховується за рахунок комбінування інтенсивності яскравості кольору та його дисперсії в спектрі різних довжин хвиль;
- lightness: схоже на значення, інтенсивність або яскравість кольору. Всі ці терміни представляють, наскільки темний / яскравий колір, однак кожна система обчислюється по-різному і

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23



використовується для різних цілей. Поточний проект не використовує жодної з цих систем, тому різниця між їхніми методами розрахунку не документується.

Причина того, що saturation та lightness (в rgb системі) не використовується в обчисленні полягає в тому, що в реальному світі (і в фотореалістичному оточенні яке використовує PBR) ідеальний колір не існує, тобто немає точно чорного  $\text{rgb}(0,0,0)$  білого  $\text{rgb}(255,255,255)$  або синього  $(0, 0, 255)$  кольору і так далі.

Тим не менше, наші очі розпізнають  $\text{rgb}(38, 109, 165)$  та  $\text{rgb}(72, 162, 233)$  як синій, і якщо дивитись на них окремо, ми явно можемо бачити різницю між цими кольорами. Якщо дивитись на їхні значення RGB, ми можемо чітко сказати, що вони значно відрізняються, що означає, що для програми вони не мають абсолютно нічого спільного.

Після конвертації двох кольорів в HSL ми отримаємо  $\text{HSL}(206, 62.6, 39.8)$  та  $\text{HSL}(206, 78.5, 59.8)$  відповідно, що дає нам однакове значення hue (відтінок), що робить обидва ці кольори однаковими відтінками синього, незалежно від їх насиченості або яскравості.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

## 3.2 Математична постановка задачі

### 3.2.1 Задача генерації середнього кольору із альбедо-текстури

Мета даної задачі – присвоїти один середній колір для кожного асету, взятий із матеріалу асету якщо такий існує, інакше – пропустити етап генерації кольору та вимкнути кольорову фільтрацію та сортування. Дано:

- Розмір альбедо текстури в пікселях;
- Растрове зображення альбедо-текстури (використовується для генерації масиву пікселей);
- Мінімальне відхилення : використовується для того, щоб виключити всі пікселі, які відрізняються по своєму кольору менше ніж на мінімальне відхилення. Допомогає пришвидшити швидкість роботи алгоритму для текстур з низьким контрастом кольорів.

Суть задачі – отримати єдиний колір із текстури розміром  $N \times M$  пікселів, за рахунок вичислення значення кожного каналу кожного пікселя, з наступним діленням кожного каналу на загальну кількість пікселів, без врахування пікселів, що були вилучені.

### 3.2.2 Задача генерації тону із RGB-кольору

Після вирахування середнього кольору RGB-кольору асету він конвертується в Hue.

Є декілька шляхів конвертації середнього RGB-кольору в його еквівалент-тон. Метод використаний в даній програмі заключається в тому, щоб знайти максимальне значення всіх трьох каналів кольору, і основуючись на результаті, використати формулу, описану в пункті 3.4.2.

Результат – це число в діапазоні від 0 до 360 (репрезентує кут повороту значення тону на кольоровому спектрі або колесі кольорів).

### 3.3 Обґрунтування методу розв'язання

Задача 3.2.1 вирішується з використанням альбедо текстури, тому що це найближче відображення кольору асету, без необхідності додавати додаткову непотрібну інформацію до даних про асет, таку як кольоровий параметр. Замість цього використовується існуюче джерело (текстура) для автоматичного розрахунку середнього кольору який потім зберігається в бібліотеці для фільтрації та сортування.

Задача 3.2.2 була вирішена шляхом конвертації RGB в значення тону, щоб уникнути обчислення 3 різних каналів, що збільшило б час роботи алгоритму втричі. В даному випадку порівнюється лише одне значення тону, яке відображає колір так само, як і RGB (інколи навіть точніше, що пояснюється в пункті 3.1).

### 3.4 Опис методів розв'язання

#### 3.4.1 Генерація середнього кольору із альбедо-текстури

Використовуваний метод - це простий і оптимізований підхід до розрахунку середнього кольору. Основна ідея цього полягає в тому, щоб скласти кожний канал всіх пікселів і розділити їх на кількість пікселів, однак, додавши мінімальне відхилення до формули, ми виключаємо необхідність доступу до кожного пікселя, а замість цього беручи до уваги під час розрахунку лише ті пікселі, що відрізняються більше, ніж на задане значення диверсії. Це значно пришвидшує розрахунок, якщо пікселі подібні за значенням, що в кінцевому рахунку означає: чим менша контрастність текстури, тим швидше вона обраховується.

Середнє значення кожного каналу обраховується за формулою:

$$R = \sum_{n=0}^x \sum_{m=0}^y (r_{n,m}),$$

$$G = \sum_{n=0}^x \sum_{m=0}^y (g_{n,m}),$$

$$B = \sum_{n=0}^x \sum_{m=0}^y (b_{n,m})$$

де R – середнє значення червоного каналу текстури;

G – середнє значення зеленого каналу текстури;

B – середнє значення синього каналу текстури;

n – горизонтальна позиція пікселя;

m – вертикальна позиція пікселя;

x – горизонтальний розмір текстури в пікселях;

y – вертикальний розмір текстури в пікселях;

r – червоне значення пікселя;

g – зелене значення пікселя;

b – синє значення пікселя.

Після цього середні значення всіх трьох каналів додаються до нового кольору, який служить середнім кольором текстури.

### 3.4.2 Генерація тону із RGB кольору

Розрахунок відтінку кольору здійснюється шляхом перетворення його в кольорове колесо (або коло, в математичних термінах), розміщенням кожного каналу в третину колеса, а потім – змішуванням кольорів між трьома положеннями, де червоний колір знаходиться в куті 0 °, зелений в 120 ° і синій в 240 °.

Далі ми перетворюємо значення всіх каналів, змішуючи їх разом з координатою кольору на колесі, що дає нам кут, який фактично є значенням відтінку кольору.

Конвертація між RGB кольором та його координатою на колесі виконується за даною формулою:

$$H_{rgb} = \text{atan2}(\sqrt{3} \cdot (G - B), 2 \cdot R - G - B)$$

Яку можна переписати у вигляді:

$$\tan(H_{rgb}) = \frac{\sqrt{3} \cdot (G - B)}{2 \cdot R - G - B}$$

Де  $H_{rgb}$  – значення тону середнього RGB-кольору

Програмна постановка наведених формул може бути представлена у вигляді трьох можливих формул, заснованих на тому, який канал кольору містить максимальне значення, що є вирішальним фактором, яка третина колірного колеса містить кінцевий результат, що значно прискорює обчислення відтінку кольору.

Формули наступні:

$$H = 60 \cdot \frac{G - B}{\max_{rgb} - \min_{rgb}} \quad \text{де } \max_{rgb} = R,$$

$$H = 60 \cdot \left( 2 + \frac{B - R}{\max_{rgb} - \min_{rgb}} \right) \quad \text{де } \max_{rgb} = G,$$

$$H = 60 \cdot \left( 4 + \frac{R - G}{\max_{rgb} - \min_{rgb}} \right) \quad \text{де } \max_{rgb} = B$$

### Висновок до розділу

В даному розділі був сформульований алгоритм конвертації альбедо-текстури, яка представляє головну текстуру асету, в середній колір цього асету. Були описані задача та ціль алгоритму. Був наведений приклад, який обґрунтовує вибір даного алгоритму.

Був сформульований алгоритм конвертації середнього кольору в його значення тону. Були описані задача, ціль та обґрунтований вибір даного алгоритму.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

В якості основного засобу розробки був обраний WPF [14] (Windows Presentation Foundation) та Visual Studio 2015 Community в якості компілятора.

WPF надає потужні засоби розробки застосувань та кастомізації екранних форм. XAML використовується як основна мова для динамічного та оптимізованого рендеру інтерфейсу. В якості основної мови програмування обраний C#.

Програма потребує швидкого та динамічного підвантаження картинок, щоб якісно виконувати функції пошуку в бібліотеці – XAML прекрасно справляється з даним завданням. Також важливо мати сучасний та зручний інтерфейс, де завантаження великої кількості асетів для відображення повинно бути виконано оптимізованим шляхом для плавної роботи програми. Також гарний інтерфейс важливий для візуального сприйняття програми, що робить пошук асетів по бібліотеці більш зручним.

Для зберігання даних бібліотеки використовується база даних Microsoft Access. Дана технологія доволі легка у використанні та швидко інтегрується в програми для Windows. В якості агента баз даних використовується Microsoft Ace Oledb 12 Database Engine [15], який працює на 64-бітних системах (що є важливим, якщо розробник працює з ігровим редактором створеним під 64-бітні системи). Також даний агент дозволяє використовувати SQL команди для швидкого пошуку по базі даних будь-яких розмірів.

Unreal Engine 4 був обраний для роботи з бібліотекою завдяки його високоякісним технологіям по створенню 3D-оточень і навіть цілих ігрових проектів в дуже короткі терміни. Також розробник на UE4 має можливість швидко створювати певні елементи функціоналу гри для тестування графіки

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

завдяки візуальній мові програмування (Unreal Blueprints). Також даний редактор має відкритий код та можливість швидкого створення та інтеграції плагінів. Кожен користувач може отримати доступ до коду редактора, редагувати будь-яку частину його функціоналу – додавання нових елементів до інтерфейсу редактора, зміна самого інтерфейсу та вбудованих функцій на C++ або Unreal Blueprints. Даний редактор можна легко зєднати з другим програмним забезпеченням, що робить можливим, наприклад, автоматизацію етапу створення асетів.

#### 4.2 Вимоги до технічного забезпечення

Для нормальної роботи програми компютер має задовольняти таким вимогам:

- комп'ютер з такою конфігурацією:
  - 1) двох ядерний процесор з тактовою частотою не нижче 1 ГГц;
  - 2) об'єм оперативної пам'яті не менше 2 ГБ.
- комп'ютерна периферія, до складу якої входить:
  - 1) монітор;
  - 2) мишка;
  - 3) клавіатура.
- додатково має бути встановлене таке програмне забезпечення:
  - 1) Операційна система Windows 7 64-bit або новіше;
  - 2) Unreal Engine 4.21 або вище;
  - 3) DirectX 11 або вище.

*деякі функції даної програми вимагають підключення до Unreal Engine 4, який має 64-бітну архітектуру. Це означає, що зєднання може бути встановлене лише якщо бібліотека також працює на 64-бітній системі.*

### 4.3 Архітектура програмного забезпечення

#### 4.3.1 Діаграма класів

Функціонування бібліотеки починається з спроби відобразити всі імпортовані асети, якщо такі є, а потім їх іконки у вікні пошуку.

Користувач може запустити на виконання одну з наступних функцій:

- пошук потрібних асетів в бібліотеці з можливістю сортування та фільтрації;
- імпорт нового асету: створюється новий об'єкт класу Asset та додається запис до бази даних. Об'єкти є повністю динамічними та створюються лише за необхідності (якщо в вікні пошуку 20 асетів – лише 20 об'єктів створено). В цей час записи в базі є статичними і використовуються для зберігання інформації про асет. Ці дані можуть бути «витягнуті» з бази за необхідності, та відфільтровані/відсортовані більш оптимальним шляхом ніж об'єкти в оперативній пам'яті;
- експорт обраних асетів: сюди входить можливість змінити налаштування експорту згідно з вимогами проекту. Один екземпляр Unreal Engine 4 має бути запущений щоб система експорту працювала коректно. Програма, створена в даній роботі, шукає шлях відкритого проекту UE4, та посилає файли асетів напряму до редактора, без необхідності імпортувати багато файлів вручну (налаштування експорту автоматично передаються від бібліотеки до ігрового редактора).

Також дане застосування виконує ряд фонових задач, в процесі імпорту, експорту та пошуку асетів. Більша частина цих задач представляють собою функцію, запущену на різних потоках, що дає користувачу можливість працювати з програмою без підвисань та зупинок основного потоку.

Також користувач може змінювати велику кількість налаштування програми, які відповідають за її функціонування та зовнішній вигляд.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31



На рисунку 4.1 наведена схема структурна класів.

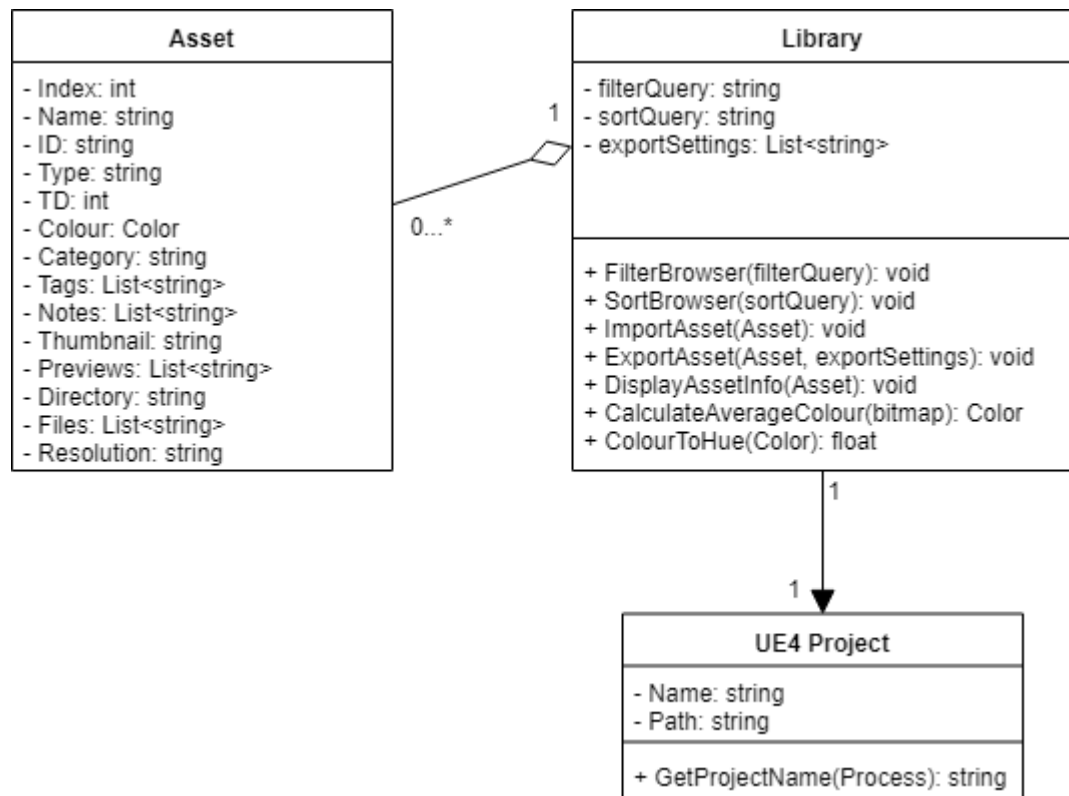


Рисунок 4.1. Діаграма класів застосування наведена в графічній частині дипломного проекту.

### 4.3.2 Діаграма послідовності

Спочатку програма читає базу даних та отримує всі доступні асети для відображення в браузері. Вона читає інформацію про асети послідовно, виходячи з їх порядку розміщення та критерію фільтрації. Також програма читає лише фіксовану кількість асетів, для більш оптимальної роботи. Кількість асетів, що читається залежить від їх іконок та позиції повзунка прокрутки. Браузер працює таким чином, що в пам'ять підвантажується інформація лише про ті асети, які відображені в вікні пошуку. Це дозволяє працювати з будь-яким розміром бази даних без підвисань програми. Тобто програма підвантажує лише ті асети, які відповідають поточній позиції повзунка прокрутки та даних фільтрації.

Під час імпорту нового асету бібліотека отримує всю необхідну інформацію із створеного об'єкта асета,

При імпорті нового асету, бібліотека отримує необхідну інформацію від створеного об'єкта, який прив'язаний до інформацію імпорту, що вводиться користувачем. Після цього програма зберігає всю інформацію до бази даних та оновлює браузер (вікно пошуку) відповідно до оновленої колекції асетів.

Під час експорту, та сама інформація, що була додана до бази під час імпорту, використовується для відправки файлів до Unreal Engine 4. Це автоматизує процес створення асетів в середині проекту за рахунок швидкого та ефективного їх експорту із бібліотеки до редактора.

На рисунку 4.2 наведена схема структурна послідовності.

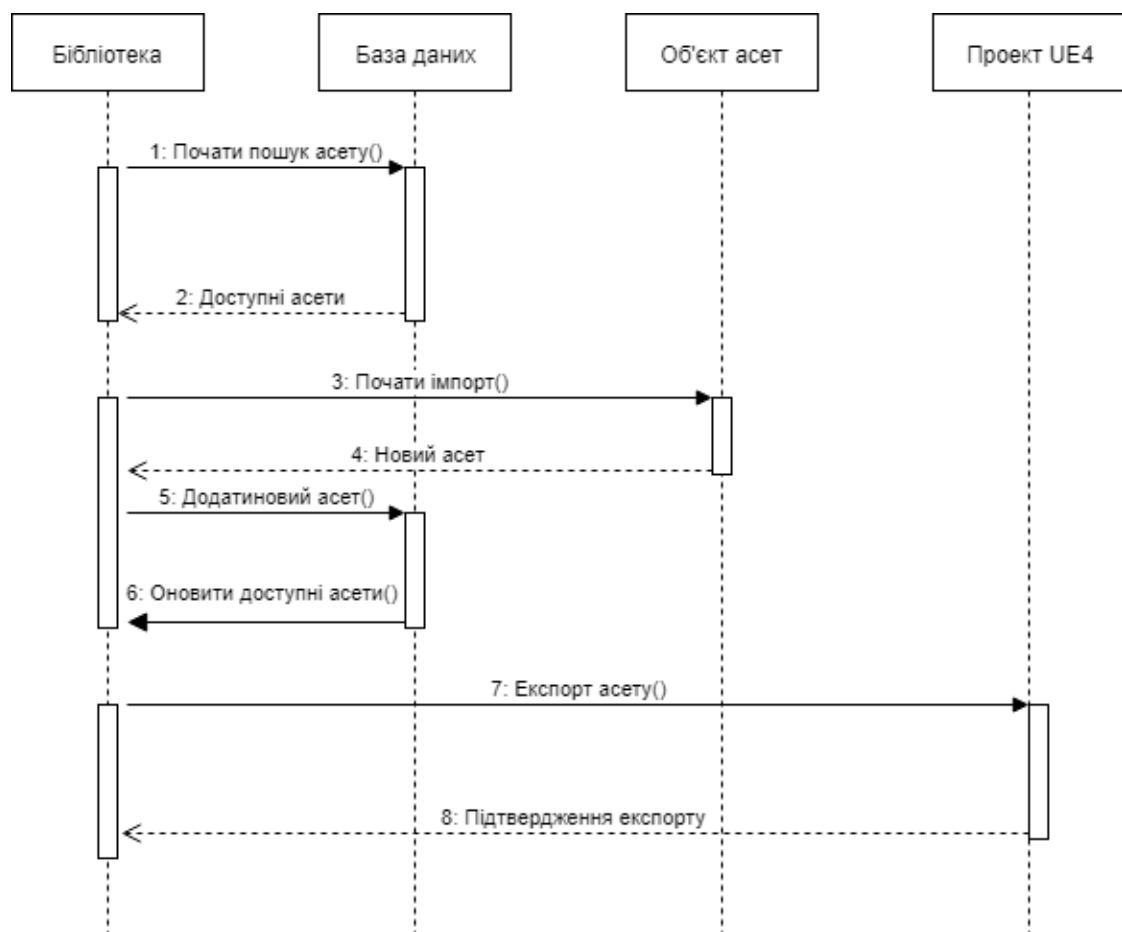


Рисунок 4.2. Схема структурна послідовності роботи з асетом наведена в графічній частині матеріалу.

### 4.3.3 Діаграма компонентів

В представленій системі є декілька ключових компонентів. На діаграм представлено 5 компонентів програми та компонент який представляє запусканий процес Unreal Engine 4.

Головний компонент застосування це Бібліотека, яка має всю інформацію про доступні асети та займається їх структуруванням та категоризацією, автоматизує різні фази на протязі життєвого циклу асета.

Компонент Браузер відповідає за читання бази даних і відображення всієї доступної інформації про асет, такої як іконка, яка зберігає інформацію про зовнішній вигляд асета. Даний компонент відправляє інформацію в бібліотеку бібліотеку, після чого вона стане доступна для інших компонентів.

Вікно інформації про асет теж є компонентом програми та зберігає специфічні деталі та автоматично отримані результати по кожному асету, який був обраний для перегляду в бібліотеці.

Вікно імпорту відповідає за надання користувачу всіх можливих засобів для додавання нового асета в бібліотеку та базу даних, для введення всієї необхідної інформації про асет, яка буде збережена в базі.

Вікно експорту дає користувачу можливість ввести власні налаштування для експорту асета. Компонент не відповідає за експорт, він лише повідомляє бібліотеці введені користувачем налаштування, які бібліотека використовує для автоматизації деяких аспектів експорту, редагування асетів, якщо потрібно, та надсилання їх в останній компонент – компонент проекту Unreal Engine 4, який отримає асети та додасть їх до проекту.

Компонент проекту Unreal Engine 4 це запусканий процес редактора, він не відноситься до головної структури даного застосування, що робить з'єднання з ним одностороннім. Таким чином UE4 лише отримує інформацію, але не відправляє жодної інформації до бібліотеки, окрім повідомлення про успішний експорт.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

На рисунку 4.3 наведена схема структурна компонентів.

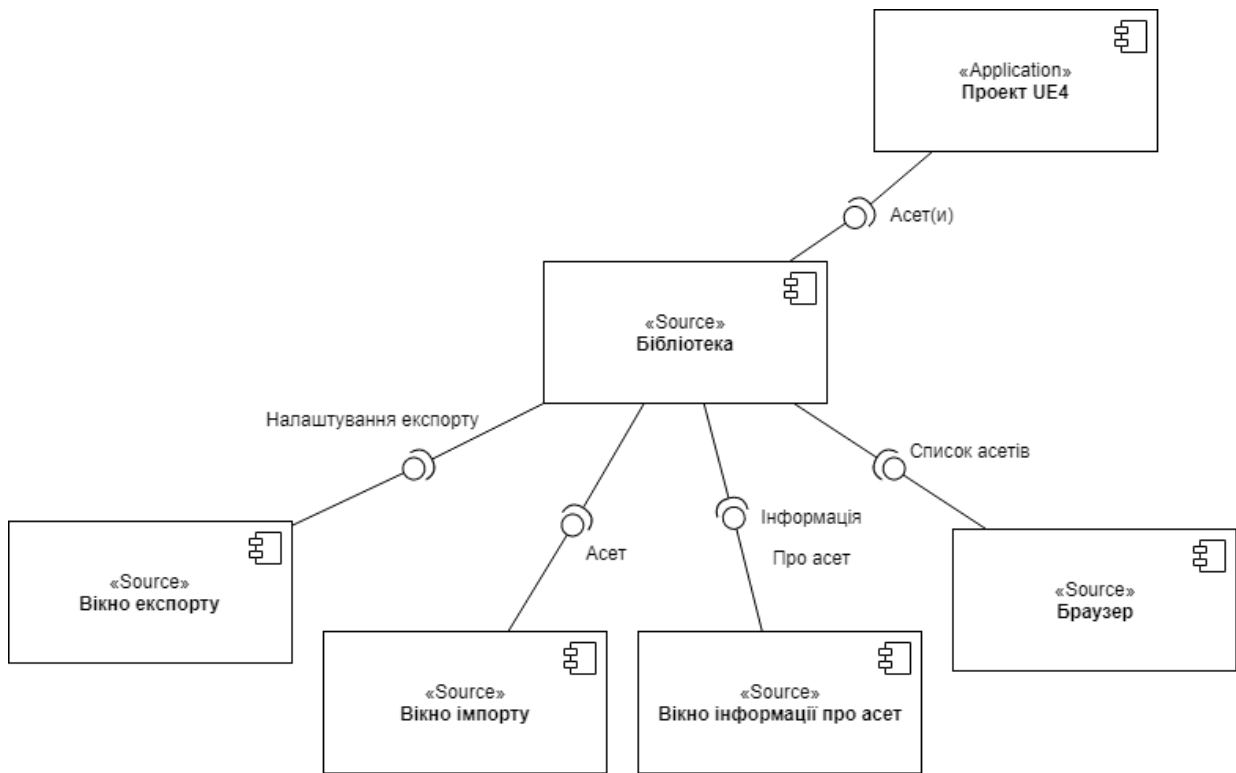


Рисунок 4.3. Схема структурна компонентів бібліотеки 3D-асетів наведена в графічному матеріалі дипломного проекту.

4.3.4 Специфікація функцій

Нижче наведена таблиця головних функцій, які відповідають за роботу програми.

Велика кількість функцій не включена в специфікацію, так як вони виконують прості логічні та арифметичні функції для підрахунку, або відповідають за роботу графічного інтерфейсу.

в таблиці 4.1 описані головні методи програмі.

Таблиця 4.1 – Опис головних методів

<i>Method</i>	<i>Дія</i>
DisplayAssets()	Відображає всі асети в базі.
FilterBrowser()	Фільтрує та сортує асети в вікні пошуку згідно з заданими критеріями.
AddThumbnail()	Викликаєбся методом <i>DisplayAssets()</i> , проходить по всім асетам та відображає їх іконку
CalculateAverageColour()	Вираховує середній колір асету
ColourToHue()	Конвертує RGB в Hue
SyncAssets()	Синхронізує асети в базі даних та в локальній бібліотеці
ReimportAllAssets()	Видаляє та заново імпортує всі асети
ImportAsset()	Додає новий асет в бібліотеку та базу даних
GetProjectName()	Знаходить ім'я поточного відкритого проекту Unreal Engine 4 та шлях до цього проекту
ExportAsset()	Експортує виділені асети до UE4-проекту

### Висновок до розділу

В даному розділі були описані програмні засоби, які використовувались для розробки програмного забезпечення, були описані їх можливості та переваги. Були наведені вимоги до системи, в якій буде працювати програма, наведені діаграми, які описують функціонування програми.

Була наведена діаграма класів, яка описує кожен клас програми.

Була наведена діаграма послідовності, яка описує процес роботи програми.

Була наведена діаграма компонентів, яка показує основні компоненти програми, їх функції та взаємозв'язки.

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

В процесі розробки програми особлива увага приділялась трьом основним функціям, які будуть детально задокументовані. Сюди входить:

- пошук. Включає в себе динамічне відображення всіх асетів, що дає користувачу можливість сортувати та фільтрувати їх основуючись на їх типі, густині текселя, кольорі, категорії, ключових словах, імені та тегах. Також пошук дає можливість сортувати асети за датою імпорту, кольором та іменем в зростаючому або спадаючому порядку. Користувач також може переглянути деталі кожного обраного асету та відкрити його директорію прямо в програмі;
- імпорт. Надає користувачу можливість додавати власні асети до бібліотеки, що є дуже важливою фішкою програми. Ігрова індустрія розвивається дуже швидко, так як і технології створення асетів. Тому дуже важливо мати можливість швидко видаляти старі та імпортувати нові асети до бібліотеки, з якою працюють розробники;
- експорт. Дана функція дозволяє економити велику частину робочого часу розробника, завдяки розширеним можливостям автоматичного експорту асетів та їх повній готовності до використання одразу після експорту в Unreal Engine 4.

5.1.1 Пошук

Після зчитування бази даних на етапі запуску програми, вона додає масив асетів до пошукового вікна, використовуючи окремий потік для уникнення зависання програми і надання користувачу можливості працювати поки зчитування відбувається.

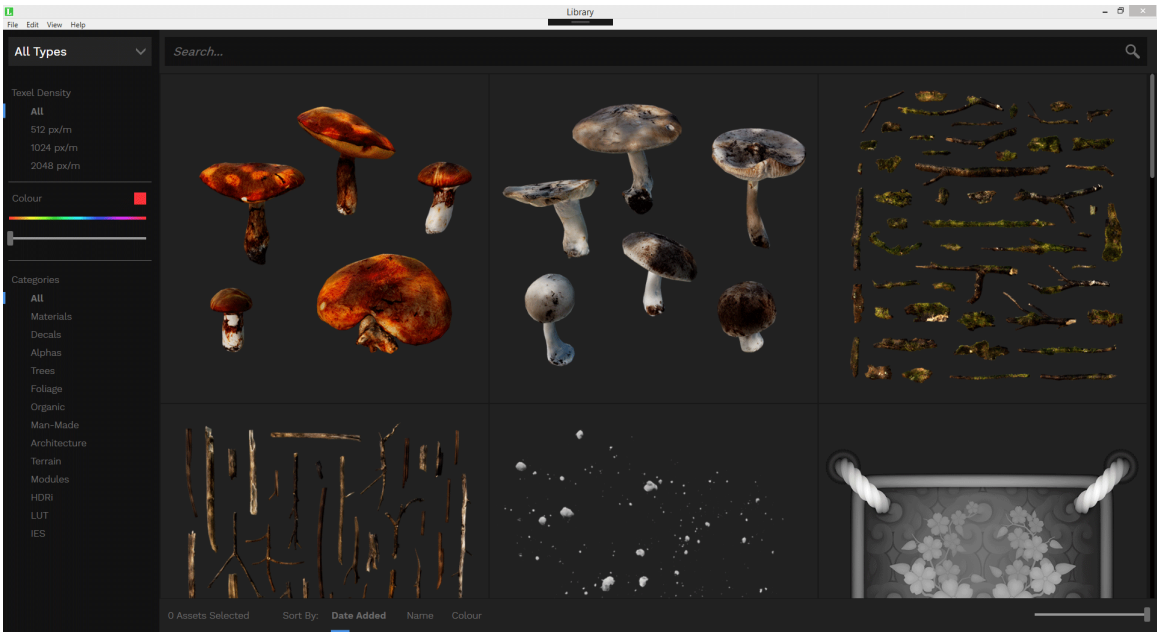


Рисунок 5.1 – Асети завантажені в пошукове вікно

Дизайн застосування виконаний в темному мінімалістичному віддтінку, щоб фокус користувача був направлений на іконки асетів, а не на яскравий інтерфейс. Це робить навігацію по бібліотеці більш зручною.

Користовуч може виділити один або декілька асетів, після цього стає активною кнопка експорту та відображається кількість обраних асетів.

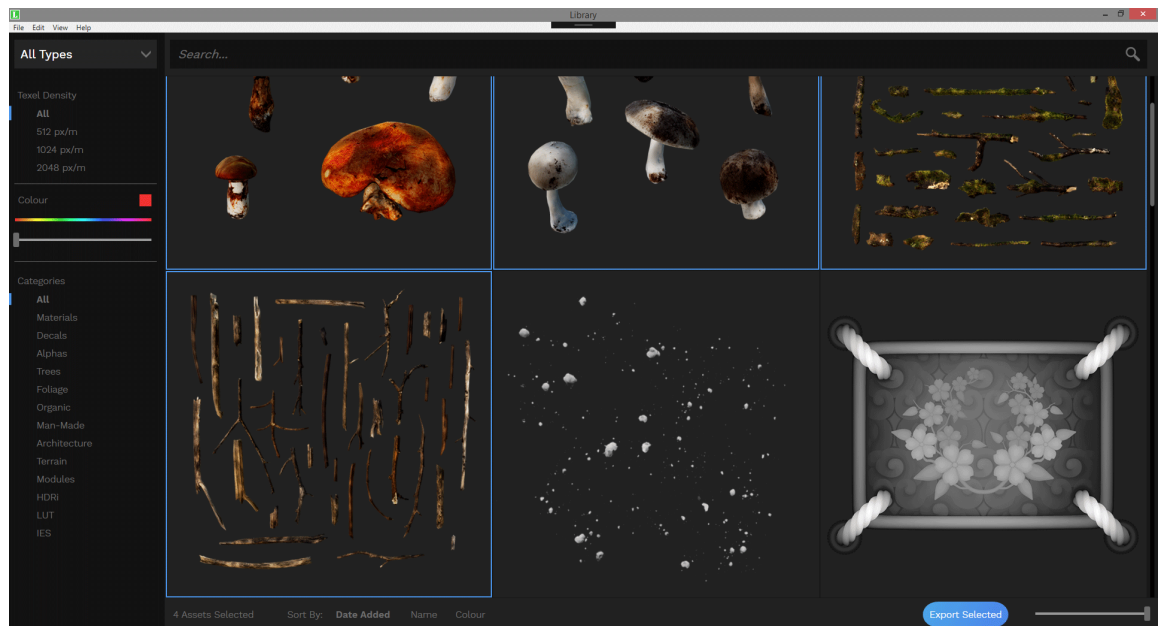


Рисунок 5.2 – Асети виділені в браузері

Також браузер має слайдер, який змінює розмір іконок, що спрощує процес експорту великої кількості асетів.

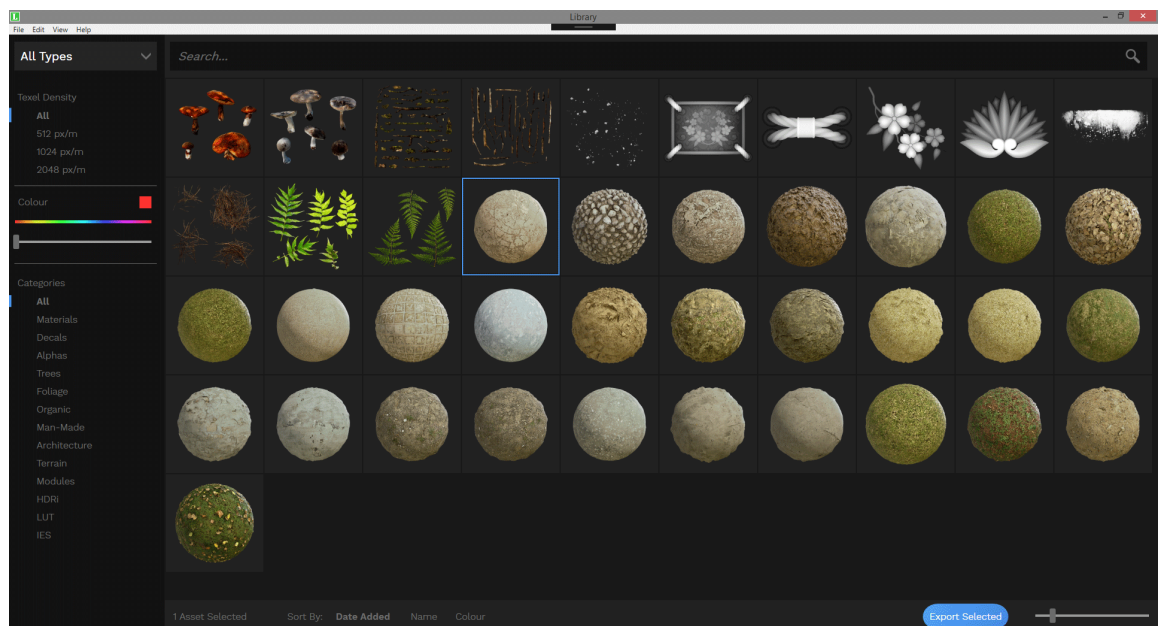


Рисунок 5.3 – Розмір іконок змінений

Зміна розміру є динамічною та проходить в реальному часі (дякуючи широким можливостям WPF та підвантаженню асетів в окремому потоці).

Під час зміни розміру іконок або прокрутки вниз, наступна група асетів підвантажується та додається до браузера. Це робить пошук оптимізованим за рахунок відсутності необхідності підвантажувати всю бібліотеку за раз.



Кожен асет може бути відкрий подвійним кліком на іконку, як результат – відображається детальна інформація про асет.

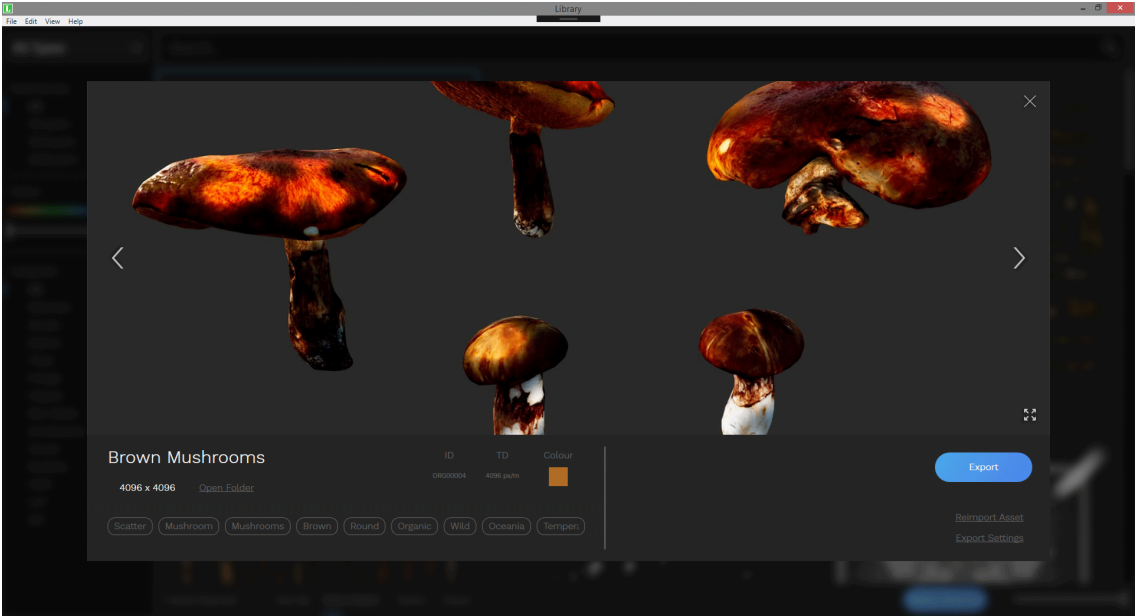


Рисунок 5.4 – Інформація про асет (модель гриба)

Користувач бачить всю потрібну інформацію та отримує доступ до налаштувань експорту та відкриття директорії асета у Windows Explorer. Натискання на тег в цьому вікні розпочне пошук з новим тегом доданим до пошукових налаштувань.

Асети можуть бути відфільтровані та відсортовані за багатьма критеріями. Навіть коли асети відфільтровані, програма все одно завантажує в пам'ять інформацію лише про відображувані асети, що значно оптимізує процес пошуку.

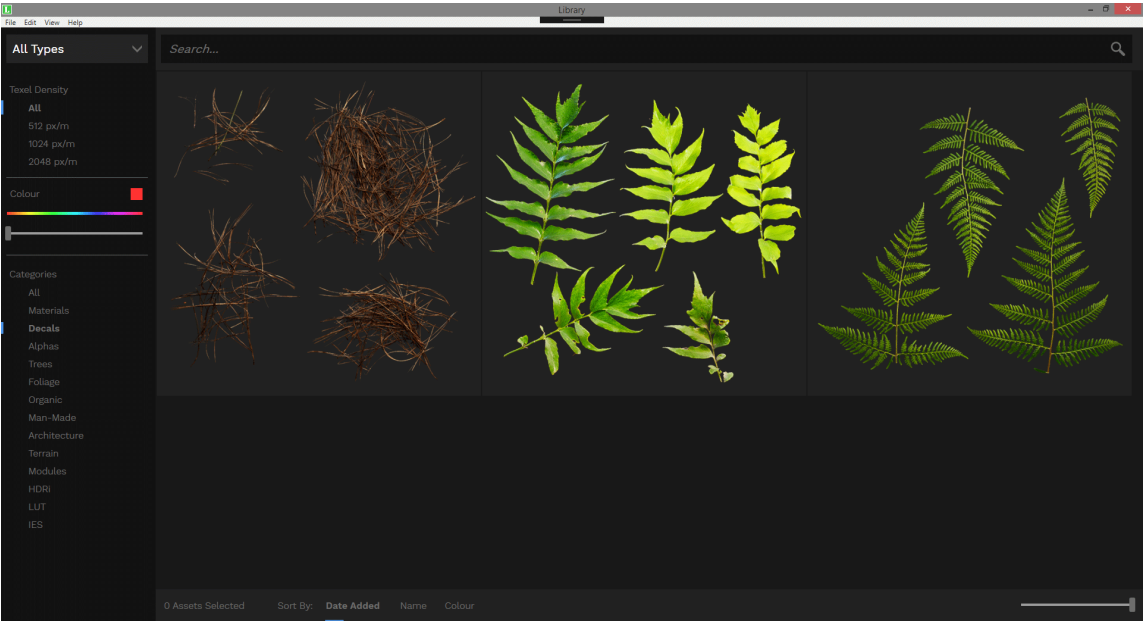


Рисунок 5.5 – Асети відфільтровані за категорією «декалі»



Рисунок 5.6 – Асети відфільтровані за зеленим кольором

Користувач може виконати додаткову фільтрацію за рахунок додавання ключових слів в пошукову строку.

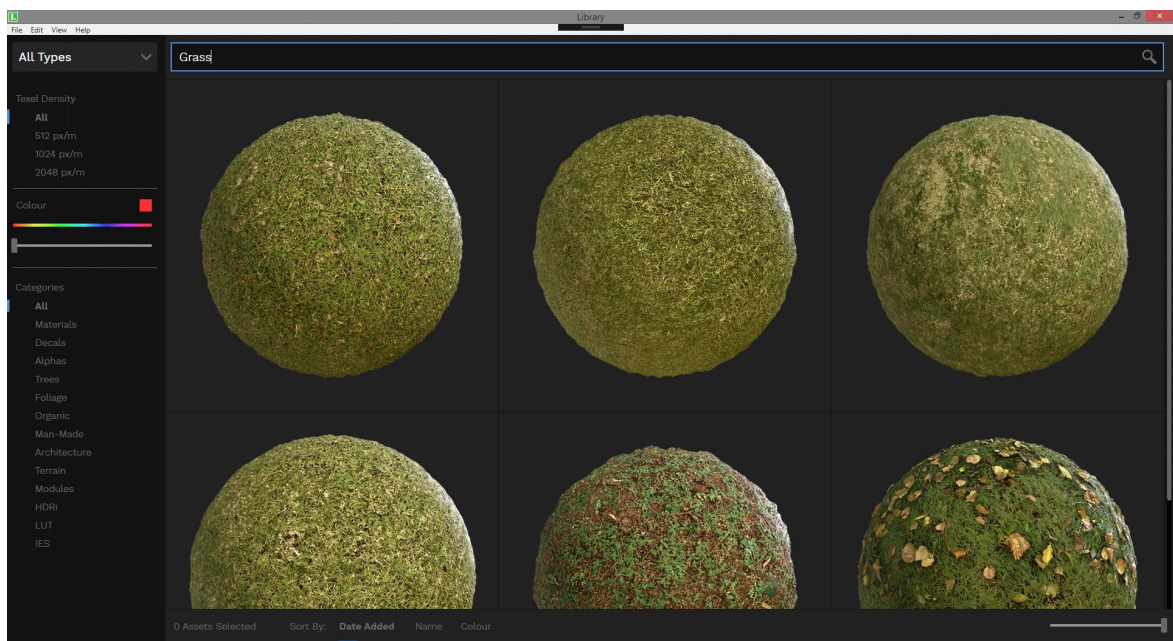


Рисунок 5.7 – Асети відфільтровані за ключовим словом «трава»

5.1.2 Імпорт

Додавання нового асету в бібліотеку виконується через *File -> Import Asset...* або комбінацією *Ctrl+N*, що відкриває вікно Windows Explorer для пошуку файлів.

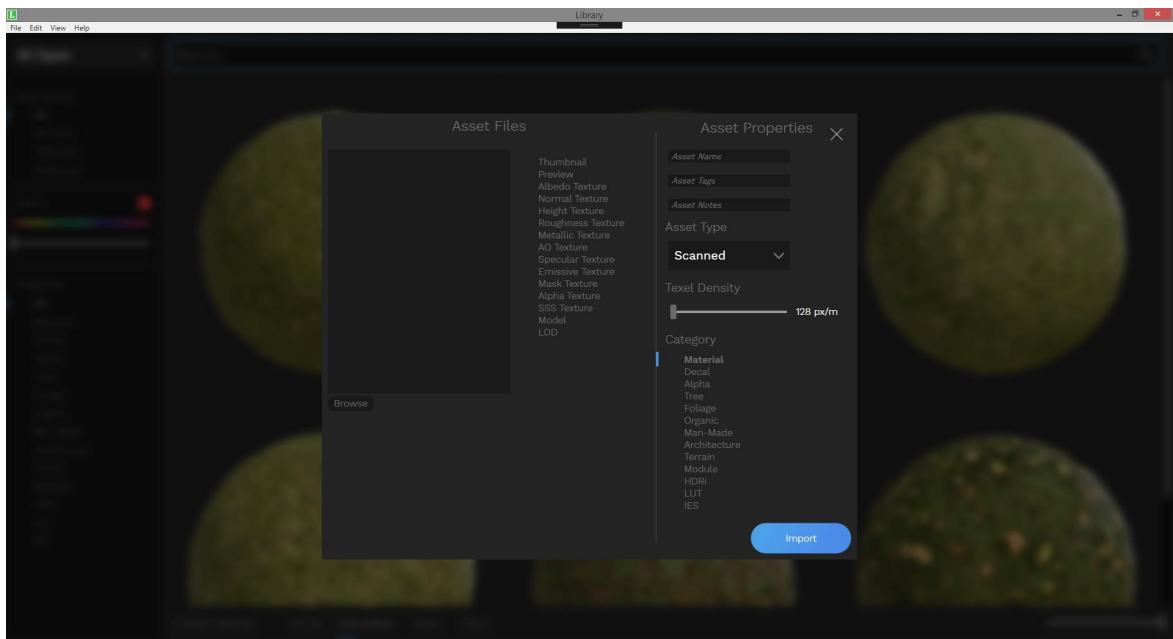


Рисунок 5.8 – Пусте вікно імпорту

Програма розпізнає наступні типи файлів:

- зображення: PNG, Jpeg, TIFF, Targa, HDRi;
- 3D-моделі: FBX, OBJ;
- інші: IES.

Після формування списку асетів, кожному файлу присвоюється тип файлу, що вказує програмі на те, як файл треба обробити. Прев'ю-зображення картинок відображається внизу вікна щоб допомогти користувачу вказати, яка текстура має який тип.

Типи текстур (картинок, зображень) наступні:

- Thumbnail\*: іконка для браузера;
- Preview: додаткове зображення, яке відображається в вікні інформації про асет. Можна обрати декілька таких зображень або жодне;
- Альbedo-текстура: використовується як “Base Colour Input” для матеріалу асета. Дана текстура використовується для генерації середнього кольору асета, що означає, в разві відсутності даної текстури асет не буде мати автоматично згенерованого середнього кольору та не буде присутнім в фільтрації/сортуванні за кольором; [16]
- Текстура нормалі: карта нормалей це текстура, яка використовується для запікання світла на сцені та відповідає за додавання мікро-деталей до поверхонь. Дана текстура додає відступ для кутів відбиття світла від поверхні матеріалу, що робить модель більш деталізованою без необхідності додавати нові полігони. Дана текстура використовує червоний та зелений канали растрового зображення як вектори для обрахунку освітлення, та синій канал інколи використовується для надання додаткового зсуву. Але синій канал рідко використовується і може бути замінений додатковою чорно-білою текстурою для економії пам'яті; [17]
- Текстура висоти: використовується як фейкова висота моделі, де яскраві пікселі – це «високо», а темні – «низько». Карта висот

використовується в багатьох випадках, наприклад при створення тесселяції (додаткові полігони для моделі) або відступ вершин моделі. Ще один метод – використання Parallax Occlusion Mapping, який надає відступ глибині моделі виходячи з пікселей текстури. Це дає моделі фейкову глибину без необхідності редагувати її геометрію; [18]

- Roughness-текстура (грубість, шорсткість): Симулюється в PBR [20] (Physically Based Rendering) щоб показати наскільки грубим є матеріал (так, наприклад, високу грубість має матеріал асфальту або дешевого паперу).

Чим вище значення пікселя, тим більш грубим є матеріал, тим слабше він відбиває світло (зеркало має нульову roughness); [19]

- Metallic-текстура: надає матеріалу властивість заліза, що змінює метод відбиття світла. Metallic-карта найбільш важлива текстура в PBR, через фізико-реалістичні та точно методи рендеру, основані на об'єктах реального світу; [21]

- Ambient Occlusion текстура: АО використовується для надання фейкових тіней деяким областям моделі. Робить модель більш деталізованою, надає тінь тим частинам об'єкту, де згідно з вимогами оточення тінь повинна бути завжди; [22]

- Specular-текстура: specular (зеркальність) часто вважається протилежним терміном до Roughness (шороховатості). Вона показує наскільки матеріал є блискучим. Рідко використовується в PBR, але з допомогою даної текстури можна видалити блиск з деяких елементів моделі, де його не повинно бути; [23]

- Emissive-текстура: фейкове освітлення моделі, основане на кольорі та яскравості пікселя. Як приклад можна навести неонову лампу, де саме матеріал зробить враження того, що лампа світиться, в той час як вбудовані джерела світла в ігровому редакторі зроблять реальне освітлення; [24]

- Текстура-маска: Маски можна використовувати з різних причин і в різних методах, але в основному це сірі текстури, які надають матеріалу інформацію про змішування між двома текстурами. Де темні пікселі відповідають ще першу текстуру, яскраві за другу;
- Альфа-текстура: надає шейдеру інформацію про те, яка частина текстури є прозорою (білий піксель) або непрозорою (чорний). Яскравий приклад – гілка дерева, яка повністю намальована на одному полігоні. На неї накладає альфа-маска, де кольори гілки замінені на чорний (непрозора частина), а всі інші на білий. Таким чином текстура буде виглядати як справжня гілка на сцені; [25]
- SSS-текстура: Sub-surface scattering або поверхневе розсіювання показує колір поверхні, коли через неї проходить світло. Використовується для реалістичної симуляції проходження світла через траву, листя, тонкі поверхні (тканину, папір); [26]
- Модель: 3D-модель представлена .FBX або .OBJ файлом. Включає інформацію про вершини, полігони та їх з'єднання, UV-карту текстур, яка дозволяє коректно накладати текстури на модель, без деформацій;
- LOD: Level of Detail (рівень деталізації) та сама модель але з меншою кількістю полігонів (менш деталізована). Більш деталізована модель замінюється на менш деталізовану автоматично, коли камері на сцені віддаляється від об'єкту (немає сенсу відрисовувати 10 тисяч полігонів, коли модель займає лише декілька пікселів на моніторі – користувач просто не побачить різниці). [27]

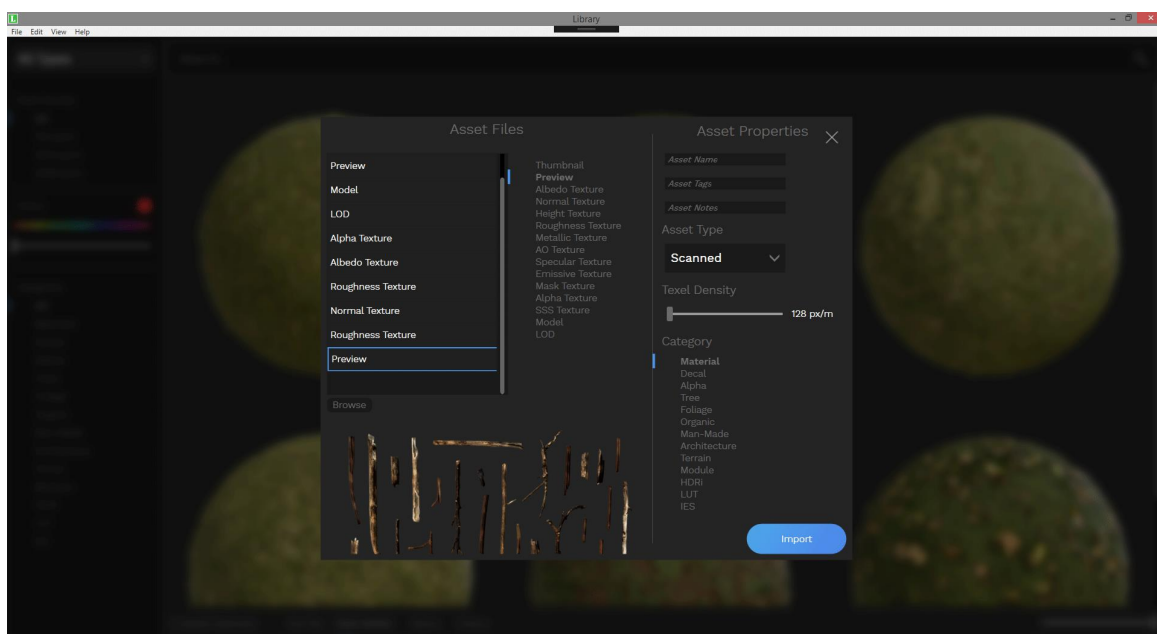


Рисунок 5.9 – Файли асету додані, тип присвоєний

Після того як всім файлам асету був присвоєний їх тип, вводиться інформація про асет (права частина вікна). Після натискання на «імпорт» асет додається до бібліотеки, зберігається в базі даних та негайно відображається в браузері. Тепер асет може бути експортований та використаний для створення оточення.

Змн.	Арк.	№ докум.	Підпис	Дата



### 5.1.3 Експорт

Функція експорту дуже проста у використанні, в тому числі завдяки автоматизації яка виконується у фоні.

Користувачу потрібно лише додати в список проектів шлях до нового проекту UE4, до якого треба робити експорт (додати проект треба лише один раз).

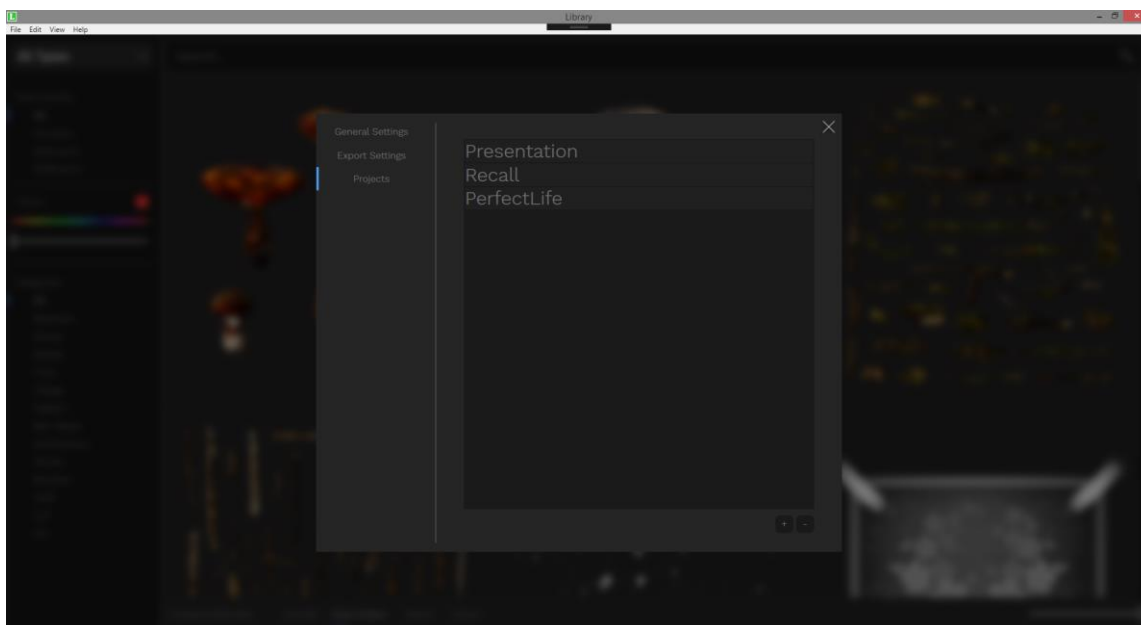


Рисунок 5.9 – Список проектів

Проекти можуть бути додані та видалені. Такий спосіб додавання проекту робить бібліотеку незалежною від нових версій UE4 – експорт буде підтримуватись завжди.

Програма перевіряє процеси що працюють, знаходить відкрите вікно редактора, порівнює ім'я процесу з іменами в списку проектів, знаходить шлях до відкритого проекту та відправляє туди асети. Експорт відбудеться автоматично виходячи з налаштувань експорту заданих користувачем.

В кінці асети з'являються в редакторі та одразу можуть бути використані для створення 3D-оточення



## 5.2 Випробування програмного продукту

Основні функції програми були перевірені на коректність:

- асети повинні коректно фільтруватись основуючись на різних критеріях фільтрації;
- асети повинні коректно сортуватись виходячи з різних налаштувань сортування;
- асети повинні коректно відображатись в браузері. Повинна з'являтись правильна іконка, а вся інформація у вікні асету повинна відповідати самому асету;
- асети повинні коректно імпортуватись в бібліотеку;
- асети повинні коректно експортуватись з бібліотеки до відкритого проекту Unreal Engine 4.

### 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач задач інформаційної підтримки створення ієрархічної структури 3D-об'єктів для 3D-сцен вимогам технічного завдання.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

В таблиці 5.1 наведені тест-кейзи для фільтрування асетів

Таблиця 5.1 – Тестування фільтрування асетів

Ситуація	Очікуваний результат
Додавання декількох ключових слів в пошук	Всі ключові слова оброблюється коректно через логічну операцію «і»
Фільтрація за категорією та ключовим словом	Асети фільтруються за обома критеріями через логічно операцію «і»
Використання спеціальних символів в пошуковій строці	Результат пошуку пустий, якщо тільки асет не має тег зі спеціальним символом в ньому

В таблиці 5.2 наведені тест-кейзи для системи сортування асетів

Таблиця 5.2 – Тестування системи сортування асетів

Ситуація	Очікуваний результат
Сортування за кольором	Асети сортуються за згенерованим середнім кольором, асети без нього не відображаються
Сортування відфільтрованих асетів	Сортування виконується, критерій фільтрації зберігається
Повторне натискання сортування	Сортування змінюється зі зростаючого на спадне і навпаки

В таблиці 5.3 наведені тест-кейзи для пошуку асетів

Таблиця 5.3 – Тестування пошуку асетів

Ситуація	Очікуваний результат
Зміна розміру іконок під час фільтрації	Іконки змінюються, можуть підвантажуватись нові асети, критерій фільтрації на місці
Подвійник клік по іконці	Відкривається вікно з інформацією про обраний асет
Подвійний клік по іконці асету, що не має превю	Іконка використовується як превю
Подвійник клік по альфа-іконца	Відкриття інформаційного вікна без полів для густини текселя або кольору

В таблиці 5.4 наведені тест-кейзи для імпорту асетів

Таблиця 5.4 – Тестування імпорту асетів

Ситуація	Очікуваний результат
Імпорт асету без іконки, тегів, імені або файлів	Користувач отримує повідомлення про недостаток інформації
Імпорт асету з некоректною альбедо-текстурою	Програма намагається вирахувати середній колір, якщо не вдалось – імпорт буде відмінений а користувач отримує повідомлення
Імпорт декількох асетів за раз	Всі асети додаються до одної папки, мають загальну іконку та ділять між собою всю іншу інформацію

В таблиці 5.5 наведені тест-кейзи експорту асетів

Таблиця 5.5 – Тестування експорту асетів

Ситуація	Очікуваний результат
----------	----------------------

Експорт файлів коли відкрита одна копія UE4	Успішний експорт до проекту
Експорт файлів коли відкрито декілька копій UE4	Користувач отримує повідомлення про те, що лише один проект має бути відкритий
Експорт файлу при закритому UE4	Користувач отримує повідомлення про те, що лише один проект має бути відкритий
Експорт файлів в проект що не був доданий до списку проектів в бібліотеці	Експорт відміняється, користувач отримує повідомлення про те, що проект треба додати в список проектів

### Висновок до розділу

В даному розділі була наведена детальна інструкція користувача для роботи з програмою. Були наведені скріншоти роботи програми та детально описаний функціонал. Був виконаний ряд тестів, включаючи стандартні та виняткові ситуації які можуть виникати при роботі з програмою.

Були описані та протестовані наступні функції програми:

- фільтрування за різними критеріями;
- сортування за різними опціями;
- комбінування фільтрації та сортування;
- пошук асетів по бібліотеці та відображення їх технічної інформації;
- імпорт нових асетів в бібліотеку та збереження інформації в базі даних;
- експорт асетів в автоматичному режимі до проекту Urneal Engine 4.

## ЗАГАЛЬНІ ВИСНОВКИ

В даній дипломній роботі під назвою «Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен», була визначені мета, цілі розробки, були описані та вирішені основні задачі розробки даної системи.

Програма була створена для того, щоб допомогти розробникам на UE4 зберегти свій час, підтримувати чисту та упорядковану структуру великої кількості 3D-асетів та з легкістю експортувати їх до проектів на UE4 для створення оточень.

Були виконані всі задачі та досягнута мета розробки. Було додано декілька задач, які потребують вирішення в наступних версіях програми для покращення її функціонування та відповідності розвитку ігрової індустрії.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

## ПЕРЕЛІК ПОСИЛАНЬ

1. Investing in the Soaring Popularity of Gaming [Електронний ресурс] – Режим доступу до ресурсу: <https://www.reuters.com/sponsored/article/popularity-of-gaming;>
2. Grand Theft Auto V [Електронний ресурс] – Режим доступу до ресурсу: [https://www.rockstargames.com/V/;](https://www.rockstargames.com/V/)
3. Rockstar Games [Електронний ресурс] – Режим доступу до ресурсу: [https://www.rockstargames.com/;](https://www.rockstargames.com/)
4. Unreal Engine 4 [Електронний ресурс] – Режим доступу до ресурсу: [https://www.unrealengine.com/en-US/;](https://www.unrealengine.com/en-US/)
5. Игровой ассет [Електронний ресурс] – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9\\_%D0%B0%D1%81%D1%81%D0%B5%D1%82;](https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B0%D1%81%D1%81%D0%B5%D1%82;)
6. Texel Density [Електронний ресурс] – Режим доступу до ресурсу: [https://80.lv/articles/textel-density-tutorial/;](https://80.lv/articles/textel-density-tutorial/)
7. Unreal Marketplace [Електронний ресурс] – Режим доступу до ресурсу: <https://www.unrealengine.com/marketplace/en-US/store;>
8. Epic Games [Електронний ресурс] – Режим доступу до ресурсу: <https://www.epicgames.com/site/en-US/home;>
9. CGTrader [Електронний ресурс] – Режим доступу до ресурсу: [https://www.cgtrader.com/;](https://www.cgtrader.com/)
10. Turbosquid [Електронний ресурс] – Режим доступу до ресурсу: [https://www.turbosquid.com/;](https://www.turbosquid.com/)
11. HDRi [Електронний ресурс] – Режим доступу до ресурсу: [https://vrender.com/what-is-hdri/;](https://vrender.com/what-is-hdri/)
12. LUT [Електронний ресурс] – Режим доступу до ресурсу: <https://www.colorgradingcentral.com/free-luts;>
13. IES [Електронний ресурс] – Режим доступу до ресурсу: [http://www.cgarena.com/freestuff/tutorials/max/ieslights/;](http://www.cgarena.com/freestuff/tutorials/max/ieslights/)

14. Windows Presentation Foundation[Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/introduction-to-wpf-in-vs>;

15. Microsoft.ACE.OLEDB.12.0 Download [Електронний ресурс] – Режим доступу до ресурсу: <https://www.microsoft.com/en-us/download/details.aspx?id=13255>;

16. Albedo Texture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cgdirector.com/albedo-map/>;

17. Normal Mapping [Електронний ресурс] – Режим доступу до ресурсу: <https://learnopengl.com/Advanced-Lighting/Normal-Mapping>;

18. Parallax Occlusion Mapping[Електронний ресурс] – Режим доступу до ресурсу: <https://learnopengl.com/Advanced-Lighting/Parallax-Mapping>;

19. Roughness Texture [Електронний ресурс] – Режим доступу до ресурсу: [https://docs.unrealengine.com/en-US/Resources/ContentExamples/MaterialNodes/1\\_4/index.html](https://docs.unrealengine.com/en-US/Resources/ContentExamples/MaterialNodes/1_4/index.html);

20. Physically-Based Rendering [Електронний ресурс] – Режим доступу до ресурсу: <https://learnopengl.com/PBR/Theory>;

21. Metalness Texture[Електронний ресурс] – Режим доступу до ресурсу: <https://www.chaosgroup.com/blog/understanding-metalness>;

22. Ambient Occlusion texture [Електронний ресурс] – Режим доступу до ресурсу: [http://wiki.polycount.com/wiki/Ambient\\_occlusion\\_map](http://wiki.polycount.com/wiki/Ambient_occlusion_map);

23. Specular Mapping [Електронний ресурс] – Режим доступу до ресурсу: <http://www.rastertek.com/dx10tut21.html>;

24. Emissive Texture [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.chaosgroup.com/display/VRAYSKETCHUP/Emissive+Material+%7C+BRDFLight>;

25. Alpha Mapping [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Alpha\\_mapping](https://en.wikipedia.org/wiki/Alpha_mapping);

26. Sub-Surface Scattering [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pluralsight.com/blog/film-games/understanding-subsurface-scattering-capturing-appearance-translucent-materials>;

27. Level Of Detail (LOD) [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.sciencedirect.com/science/article/pii/B9781558608382500030>.

					ДП ІС-5228.1260-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55



## Додаток А

**Тексти програмного коду**Інформаційна підтримка створення ієрархічної структури 3D-об'єктів

(Найменування програми (документа))

DVD-R

(Вид носія даних)

10 арк, 2930 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

					ДП ІС-5228.1260-с.ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

```

namespace EnvironmentAssetLibrary
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        #region Main Parameters
        /// <summary>
        /// Main program parameters
        /// </summary>
        public int ThumbnailSize = 500;

        List<Image> Thumbnails = new List<Image>();

        List<ListBoxItem> ThumbnailItems = new List<ListBoxItem>();

        OleDbCommand cmd = new OleDbCommand();

        OleDbConnection cn = new OleDbConnection();

        DataTable dt;

        ScrollViewer BrowserScrollbar;

        List<Asset> LoadedAssets = new List<Asset>();

        int LoadedAssetIndex = 0;

        double BrowserVerticalOffset = 0;

        double BrowserMaxVerticalOffset = 0;

```

```
public string LibraryDirectory =
"D:/Library/Assets/";
```

```
public string UserManualLink =
"https://drive.google.com/open?id=1lYZlAZxVgKxAoab_-y0sehZz-y9mm4lzGcCPk9S4V1g";
```

```
string filterCategory = "All";
```

```
string filterTD = "All";
```

```
string filterType = "All Types";
```

```
double filterColour = 500;
```

```
bool bFilterColour = false;
```

```
double filterColourRange = 10;
```

```
string filterSearch = "null";
```

```
string SortBy = " ORDER BY ID Desc";
```

```
string SortDirection = " Asc";
```

```
bool SortChanged = false;
```

```
private bool ColourSliderDragging = false;
```

```
#endregion
```

```
#region Base
```

```
/// <summary>
```

```
/// Base functions;s constructor and loading events.
```

```
/// </summary>
```

```
public MainWindow()
```

```
{
```

```
    InitializeComponent();
```

```
    Style = (Style)FindResource(typeof(Window));
```

					ДП ІС-5228.1260-с.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this.DataContext = this;

        cn.ConnectionString = @"Provider = Microsoft.ACE.OLEDB.12.0; Data Source = " +
LibraryDirectory + "AssetDB.mdb";

        cmd.Connection = cn;

        SyncAssets();

    }

private void MainLibraryWindow_ContentRendered(object sender, EventArgs e)
{
    ThumbnailSizeSlider.Value = ThumbnailSizeSlider.Maximum;

    DisplayAssets();
}

void RestartApplication()
{
    System.Diagnostics.Process.Start(Application.ResourceAssembly.Location);

    Application.Current.Shutdown();
}

#endregion

#region Database Reading

/// <summary>

/// Reading from database

/// Reading based on filters and sorting settings set in the ui

/// Assets are read one line at a time (browser window size / thumbnail size)

/// New assets are read and loaded as the scrollbar reaches the bottom of the window

/// Reading is dynamic and is not resource-dependant; the database is located in the library
path, and is generated if not found

```

/// A new asset object is added  
for each asset read from the database

/// </summary>

public void DisplayAssets()

{

int AssetIndex;

string AssetName;

string AssetID;

string AssetType;

int AssetTD;

System.Drawing.Color AssetColour;

string AssetCategory;

List<string> AssetTags;

List<string> AssetNotes;

string AssetThumbnail;

List<string> AssetPreviews;

string AssetDirectory;

List<string> AssetFiles;

string AssetResolution;

cmd.CommandText = FilterBrowser();

if (cn.State != ConnectionState.Open)

{

cn.Open();

}

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

dt = new DataTable();

					ДП ІС-5228.1260-с.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

da.Fill(dt);

for(int i = 0; i < ((int)Math.Floor(ThumbnailBrowser.ActualWidth / ThumbnailSize))*4;
i++)
{
    if (LoadedAssetIndex < dt.Rows.Count)
    {
        AssetIndex = LoadedAssetIndex;

        AssetName = String.Format(dt.Rows[LoadedAssetIndex][1].ToString());

        AssetID = String.Format(dt.Rows[LoadedAssetIndex][2].ToString());

        AssetType = String.Format(dt.Rows[LoadedAssetIndex][3].ToString());

        AssetTD = Int32.Parse(dt.Rows[LoadedAssetIndex][4].ToString());

        var splitString = String.Format(dt.Rows[LoadedAssetIndex][5].ToString()).Split(',');

        var splitInts = splitString.Select(item => int.Parse(item)).ToArray();

        AssetColour = System.Drawing.Color.FromArgb(splitInts[0], splitInts[1],
splitInts[2]);

        AssetCategory = String.Format(dt.Rows[LoadedAssetIndex][6].ToString());

        AssetTags =
String.Format(dt.Rows[LoadedAssetIndex][7].ToString()).Split('|').ToList();

        AssetNotes =
String.Format(dt.Rows[LoadedAssetIndex][8].ToString()).Split('|').ToList();

        AssetDirectory = LibraryDirectory + AssetCategory + "/" + AssetID + "/";

        AssetThumbnail = AssetDirectory + AssetID + "_Thumbnail.png";

        AssetPreviews = Directory.GetFiles(AssetDirectory, AssetID +
"_Preview*").ToList();

        AssetFiles = Directory.GetFiles(AssetDirectory, AssetID +
"_T_*").ToList().Concat(Directory.GetFiles(AssetDirectory, AssetID +
"_SM_*").ToList()).ToList();

        AssetResolution = String.Format(dt.Rows[LoadedAssetIndex][9].ToString());

```

```

        Asset asset = new
EnvironmentAssetLibrary.Asset(AssetIndex, AssetName, AssetID, AssetType, AssetTD,
AssetColour, AssetCategory, AssetTags, AssetNotes, AssetThumbnail, AssetPreviews,
AssetDirectory, AssetFiles, AssetResolution);

```

```

        LoadedAssets.Add(asset);

```

```

        LoadedAssetIndex++;

```

```

        AddThumbnail(AssetThumbnail, AssetName, AssetIndex);

```

```

    }

```

```

}

```

```

cn.Close();

```

```

if (ThumbnailBrowser.Items.Count == 0)

```

```

{

```

```

    EmptyBrowserMessage.Visibility = Visibility.Visible;

```

```

}

```

```

else

```

```

{

```

```

    EmptyBrowserMessage.Visibility = Visibility.Hidden;

```

```

}

```

```

}

```

```

List<int> GetVacantIDs()

```

```

{

```

```

    cmd.CommandText = "select * from Assets ORDER BY ID";

```

```

    if (cn.State != ConnectionState.Open)

```

```

    {

```

```

        cn.Open();

```

```

    }

```

```

OleDbDataAdapter da = new
OleDbDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

List<int> idIndex = new List<int>();

int missingCount = 0;

if(dt.Rows.Count != 0)
{
    for (int i = 0; i < (int)dt.Rows[dt.Rows.Count - 1][0]; i++)
    {
        if ((int)dt.Rows[missingCount][0] != i + 1)
        {
            idIndex.Add(i + 1);
        }
        else
        {
            missingCount++;
        }
    }

    cn.Close();

    idIndex.Add((int)dt.Rows[dt.Rows.Count - 1][0] + 1);

    return idIndex;
}

idIndex.Add(1);

return idIndex;
}

```



```

List<string>
RemoveAssetDuplicates(List<string> asset)
{
    cmd.CommandText = "select * from Assets ORDER BY ID";

    if (cn.State != ConnectionState.Open)
    {
        cn.Open();
    }

    OleDbDataAdapter da = new OleDbDataAdapter(cmd);

    dt = new DataTable();

    da.Fill(dt);

    List<string> availableID = new List<string>();

    for(int i = 0; i < dt.Rows.Count; i++)
    {
        availableID.Add(LibraryDirectory + dt.Rows[i][6].ToString() + @"\" +
dt.Rows[i][2].ToString() + @"\" + dt.Rows[i][2].ToString() + ".ini");
    }

    return asset.Except(availableID).ToList();
}

#endregion

```

## #region Browser Filtering

/// &lt;summary&gt;

/// Filtering happens by passing filter keys from the ui to a query that gets constructed every time filtering ui changes.

/// &lt;/summary&gt;

private string FilterBrowser()

{

string query = null;

if (filterCategory == "All")

{

query = "select \* from Assets WHERE Category IS NOT NULL";

}

else

{

query = "select \* from Assets WHERE Category = '" + filterCategory + "'";

}

if (filterTD == "All")

{

query += "";

}

else

{

query += " AND TD &gt;= " + filterTD.Substring(0, filterTD.Length - 5);

}

if(filterType == "All Types")

{

query += "";

}

					ДП ІС-5228.1260-с.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

else

{

    query += " AND Type = '" + filterType + "'";

}

if(filterColour == 500)

{

    query += "";

}

else

{

    query += " AND Colour <> '0,0,0'" + " AND Hue BETWEEN " + (filterColour -
filterColourRange) + " AND " + (filterColour + filterColourRange);

}

if(filterSearch != "null")

{

    string[] keyWords = filterSearch.Split(' ');

    foreach(string word in keyWords)

    {

        query += " AND (Tags LIKE '%" + word + "%' OR AssetName LIKE '%" + word +
"%')";

    }

}

query += SortBy;

return query;

}

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис)      С.Л.Проскура  
(ініціали, прізвище)

“17” квітня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_  
(підпис)      О.А.Павлов  
(ініціали, прізвище)

“18” квітня 2019 р.

Інформаційна підтримка створення ієрархічної структури 3D-  
об’єктів для 3D-сцен

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр ДП ІС-5228.1260-с.ТЗ

На 10 сторінках

Київ – 2019 року

# ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	2
1.1	Повне найменування системи та її умовне позначення .....	2
1.2	Найменування організації-замовника та організації-учасника робіт.....	2
1.3	Перелік документів, на підставі яких створюється система .....	2
1.4	Планові терміни початку і закінчення роботи зі створення системи.....	3
2	ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ КОМПЛЕКСУ ЗАДАЧ .....	4
2.1	Призначення комплексу задач .....	4
2.2	Цілі створення комплексу задач.....	4
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	6
4.1	Вимоги до функціональних характеристик .....	6
4.2	Вимоги до надійності .....	7
4.3	Вимоги до складу і параметрів технічних засобів .....	7
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ .....	9
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ.....	10
6.1	Види випробувань.....	10

					<b>ДП ІС-5228.1260-с.ТЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Інформаційна підтримка створення ієрархічної структури 3D об'єктів для 3D- сцен</i>	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розроб.</i>		<i>Юсеф А.</i>						
<i>Перевірів.</i>		<i>Проскура С.Л.</i>					2	10
						<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351</i>		
<i>Н. кон.</i>		<i>Телишева Т.О.</i>						
<i>Затв.</i>		<i>Проскура С.Л.</i>						

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Повне найменування системи та її умовне позначення

Повна назва системи: *Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен*

## 1.2 Найменування організації-замовника та організації-учасника робіт

Генеральним замовником проекту являється кафедра Автоматизованих систем обробки інформації та управління НТУУ "КПІ". Представником замовника є Проскура Світлана Леонідівна.

Розробником системи є студент групи ІС- 351 факультету інформатики та обчислювальної техніки НТУУ «КПІ ім. Ігоря Сікорського» Юсеф Алі.

## 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП ІС-5228.1260-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

#### 1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням системи створення ієрархічної структури 3D-об'єктів для 3D-сцен – 5 лютого 2019 рік.

Плановий термін по закінченню роботи над створенням системи створення ієрархічної структури 3D-об'єктів для 3D-сцен – не пізніше 1 червня 2019 року.

					ДП ІС-5228.1260-с.ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ КОМПЛЕКСУ ЗАДАЧ

### 2.1 Призначення комплексу задач

Призначенням розробки є організація, фільтрування та сортування 3D-асетів для збереження в бібліотеці та для використання при створенні 3D-середовища.

### 2.2 Цілі створення комплексу задач

Цілями розробки є спрощення процесу роботи дизайнера 3D-середовища, який працює з Unreal Engine 4, за рахунок:

- створення структурованої бібліотеки асетів зі своїми категоріями;
- зменшення часу, який витрачається на пошук та імпорт асетів в Unreal Engine 4 завдяки автоматизації деяких частин процесу створення та експорту;
- надання можливості підтримувати велику та динамічну бібліотеку асетів, яку можна використовувати для розробки різних проектів та 3D-середовищ.

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- розробити підсистему для пошуку, сортування, та фільтрації асетів за певними критеріями;
- розробити підсистему отримання інформації про асети;
- розробити базу даних поточних асетів для розширення бібліотеки;
- розробити підсистему експорту асетів до Unreal Engine 4;
- розробити підсистему імпорту нових асетів до бібліотеки.

					ДП ІС-5228.1260-с.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		



### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Всі розробники на Unreal Engine 4 можуть використовувати дану бібліотеку щоб додавати асети до своїх проектів. В процесі використання користувач може легко та швидко вибрати необхідні асети для проекту, імпортувати їх для використання при розробці 3D-середовищ або для використання в готовому середовищі.

Об'єктом автоматизації є створення критеріїв фільтрації, пакування текстур в пресет, конвертування текстур між різними форматами, зміна їх розміру, іменування за визначеними правилами.

					ДП ІС-5228.1260-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Застосування повинно надавати доступ до великої кількості категоризованих асетів та відповідати наступним вимогам:

- система надає користувачу можливість шукати потрібні асети серед всіх доступних:
  - 1) система надає можливість шукати, сортувати та фільтрувати асети;
  - 2) система має синхронізувати бібліотеку з локальними асетами, або асетами доданими в базу даних;
  - 3) система надає можливість видалити кожен асет з бібліотеки з можливістю видалення локальних файлів.
- система має забезпечити можливість імпорту асетів в бібліотеку:
  - 1) система надає можливість вказати категорію та певні параметри фільтрації нового асету;
  - 2) система автоматизує деякі частини фази імпорту (перейменування, обрання ієрархічно-вірної директорії для розміщення, автоматичне створення критеріїв фільтрації).
- система надає користувачу можливість експортувати обрані асети з можливістю налаштування параметрів експорту для кожного асету або для всіх асетів що експортуються одночасно;
- система надає користувачу можливість змінювати параметри перегляду бібліотеки, щоб зробити процес пошуку зручнішим.

					ДП ІС-5228.1260-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4.2 Вимоги до надійності

Програма повинна зберігати працездатність і забезпечувати відновлення своїх функцій при виникненні наступних позаштатних ситуацій:

- введення некоректних налаштувань імпорту;
- обрання некоректних файлів для імпорту;
- втрата бази даних або локальних файлів.

Програма має намагатись зберігати свій поточний стан в разі виникнення помилки та відновлювати будь-який пошкоджений файл в разі можливості, видаляти з бази даних входження про файли, які не вдалося відновити.

#### 4.3 Вимоги до складу і параметрів технічних засобів

Склад, структура і способи організації даних в системі повинні бути визначені на етапі технічного проектування.

Структура технічних засобів визначається виходячи із можливості їх забезпечити процедуру імпорту та перегляду щойно доданих асетів.

Для правильної роботи розробленої системи до складу технічних засобів повинен входити комп'ютер, що має конфігурацію наведену нижче:

- комп'ютер з такою конфігурацією:
  - 1) двох ядерний процесор процесор з тактовою частотою не нижче 1 ГГц;
  - 2) об'єм оперативної пам'яті не менше 2 ГБ.
- комп'ютерна периферія, до складу якої входить:
  - 1) монітор;
  - 2) мишка;
  - 3) клавіатура.

- додатково має бути встановлене таке програмне забезпечення:
  - 1) операційна система Windows 7 64-bit або новіше;
  - 2) Unreal Engine 4.21 або вище;
  - 3) DirectX 11 або вище.

					ДП ІС-5228.1260-с.ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з автоматизації формування асортименту торгівельної організації.

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	07.02.2019	
2.	Розробка сценарію роботи	12.02.2019	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	20.02.2019	
4.	Узгодження з керівником інтерфейсу користувача	02.03.2019	
5.	Розробка інформаційного забезпечення	17.03.2019	
6.	Розробка програмного забезпечення	29.03.2019	
7.	Налагодження програми	25.04.2019	
8.	Тестування програми	08.05.2019	
9.	Здача готового програмного продукту замовнику	14.05.2019	

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

### 6.1 Види випробувань

Для контролю правильності роботи програмного забезпечення буде проведено модульне тестування (Unit Test). В ході тестування буде проведено випробування основних елементів системи, таких як імпорт нових асетів та експорт обраних асетів до Unreal Engine 4, компоненти інтерфейсу користувача (UI-компоненти).

					ДП ІС-5228.1260-с.ТЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис) С.Л.Проскура  
(ініціали, прізвище)

“11” травня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувач кафедри**

\_\_\_\_\_  
(підпис) О.А. Павлов  
(ініціали, прізвище)

“12” травня 2019 р.

Інформаційна підтримка створення ієрархічної структури 3D-  
об’єктів для 3D-сцен

**ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

Шифр ДП ІС-5228.1260-с.ПМВ

на 10 сторінках

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ОБ'ЄКТ ВИПРОБУВАННЯ.....</b>	<b>3</b>
1.1	Найменування програми .....	3
1.2	Область застосування .....	4
1.3	Умовне позначення програми.....	4
<b>2</b>	<b>МЕТА ВИПРОБУВАНЬ .....</b>	<b>5</b>
<b>3</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>6</b>
3.1	Вимоги до функціональних характеристик .....	6
3.1.1	Вимоги до складу виконуваних функцій .....	6
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....</b>	<b>7</b>
<b>5</b>	<b>СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ.....</b>	<b>8</b>
<b>6</b>	<b>МЕТОДИ ВИПРОБУВАНЬ .....</b>	<b>9</b>

					ДП ІС-5228.1153-с.ПМВ							
Зм.	Арк.	Прізвище	Підпис	Дата								
Розроб.		Юсеф А.			Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен			Літ.	Лист	Листів		
										2	10	
Перевірив.		Проскура С.Л.						КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351				
Н. кон.		Телишева Т.О.										
Затв.		Проскура С.Л.										



## 1 ОБ'ЄКТ ВИПРОБУВАННЯ

### 1.1 Найменування програми

Темою дипломного проекту є «Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен» із назвою програмного продукту – Environment Asset Library.

### 1.2 Область застосування

Область застосування програми – організація бібліотеки 3D-асетів для збереження часу розробника 3D-оточень в Unreal Engine 4.

Програма автоматизує процес імпорту асетів а також підтримує структуру асетів для спрощення процесу фільтрації та пошуку.

### 1.3 Умовне позначення програми

Умовне позначення програми – WPF Application: Environment Asset Library.

## 2 МЕТА ВИПРОБУВАНЬ

Метою випробувань являється перевірка відповідності функцій комплексу задач задач інформаційної підтримки створення ієрархічної структури 3D-об'єктів для 3D-сцен вимогам технічного завдання.

					ДП ІС-5228.1260-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Вимоги до функціональних характеристик

Програмне застосування повинно:

- асети повинні коректно фільтруватись основуючись на різних критеріях фільтрації;
- асети повинні коректно сортуватись виходячи з різних налаштувань сортування;
- асети повинні коректно відображатись в браузері. Повинна з'являтись правильна іконка, а вся інформація у вікні асету повинна відповідати самому асету;
- асети повинні коректно імпортуватись в бібліотеку;
- асети повинні коректно експортуватись з бібліотеки до відкритого проекту Unreal Engine 4.

##### 3.1.1 Вимоги до складу виконуваних функцій

Застосування має виконувати наступні функції:

- автоматизувати процес імпорту;
- фільтрувати та сортувати асети;
- проводити пошук асетів основуючись на їх іменах та тегах;
- додавати, видаляти та синхронізувати асети з локальними об'єктами бібліотеки;
- автоматизувати процес експорту до Unreal Engine 4.

#### 4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмний продукти розробляється на основі Технічного Завдання.

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

					ДП ІС-5228.1260-с.ПМВ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

**Умови проведення випробувань:** працююча програма з тестовими асетами, доданими до бібліотеки.

**Умови початку та завершення окремих етапів тестування:** тестування кожних елементів програми має відбуватися з урахуванням всіх можливих виключних ситуацій в залежності від функціональних можливостей програмного продукту.

**Обмеження щодо умов проведення тестування:** тестування має проводитися в рамках функціонального апарату програмного забезпечення.

**Вимоги до технічного обслуговування системи:** система має бути пристроєм на операційній системі Windows, Unreal Engine 4 має бути встановлений, проект на Unreal Engine 4 має бути відкритий в процесі тестування.

**Міри, забезпечуючі безпеку та безаварійність проведення тестування:** тестування системи не може визвати аварійних ситуацій.

**Порядок взаємодій організацій, які беруть участь у тестуванні:** тестування проводить один студент КПП групи ІС- 351 Юсеф Алі.

					ДП ІС-5228.1260-с.ПМВ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## 6 МЕТОДИ ВИПРОБУВАНЬ

У наступних таблицях наведений перелік випробувань основних функціональних можливостей.

**Таблиця 6.1** – Тестування фільтрування асетів

Ситуація	Очікуваний результат
Додавання декількох ключових слів в пошук	Всі ключові слова оброблюється коректно через логічну операцію «і»
Фільтрація за категорією та ключовим словом	Асети фільтруються за обома критеріями через логічно операцію «і»
Використання спеціальних символів в пошуковій строці	Результат пошуку пустий, якщо тільки асет не має тег зі спеціальним символом в ньому

**Таблиця 6.2** – Тестування системи сортування асетів

Ситуація	Очікуваний результат
Сортування за кольором	Асети сортуються за згенерованим середнім кольором, асети без нього не відображаються
Сортування відфільтрованих асетів	Сортування виконується, критерій фільтрації зберігається
Повторне натискання сортування	Сортування змінюється зі зростаючого на спадне і навпаки

Таблиця 6.3 – Тестування пошуку асетів

Ситуація	Очікуваний результат
Зміна розміру іконок під час фільтрації	Іконки змінюються, можуть підвантажуватись нові асети, критерій фільтрації на місці
Подвійник клік по іконці	Відкривається вікно з інформацією про обраний асет
Подвійний клік по іконці асету, що не має превю	Іконка використовується як превю
Подвійник клік по альфа-іконця	Відкриття інформаційного вікна без полів для густини текселя або кольору

Таблиця 6.4 – Тестування імпорту асетів

Ситуація	Очікуваний результат
Імпорт асету без іконки, тегів, імені або файлів	Користувач отримує повідомлення про недостаток інформації
Імпорт асету з некоректною альбедо-текстурою	Програма намагається вирахувати середній колір, якщо не вдалось – імпорт буде відмінений а користувач отримує повідомлення
Імпорт декількох асетів за раз	Всі асети додаються до одної папки, мають загальну іконку та ділять між собою всю іншу інформацію

Таблиця 6.5 – Тестування експорту асетів

Ситуація	Очікуваний результат
Експорт файлів коли відкрита одна копія UE4	Успішний експорт до проекту
Експорт файлів коли відкрито декілька копій UE4	Користувач отримує повідомлення про те, що лише один проект має бути відкритий
Експорт файлу при закритому UE4	Користувач отримує повідомлення про те, що лише один проект має бути відкритий
Експорт файлів в проект що не був доданий до списку проектів в бібліотеці	Експорт відміняється, користувач отримує повідомлення про те, що проект треба додати в список проектів

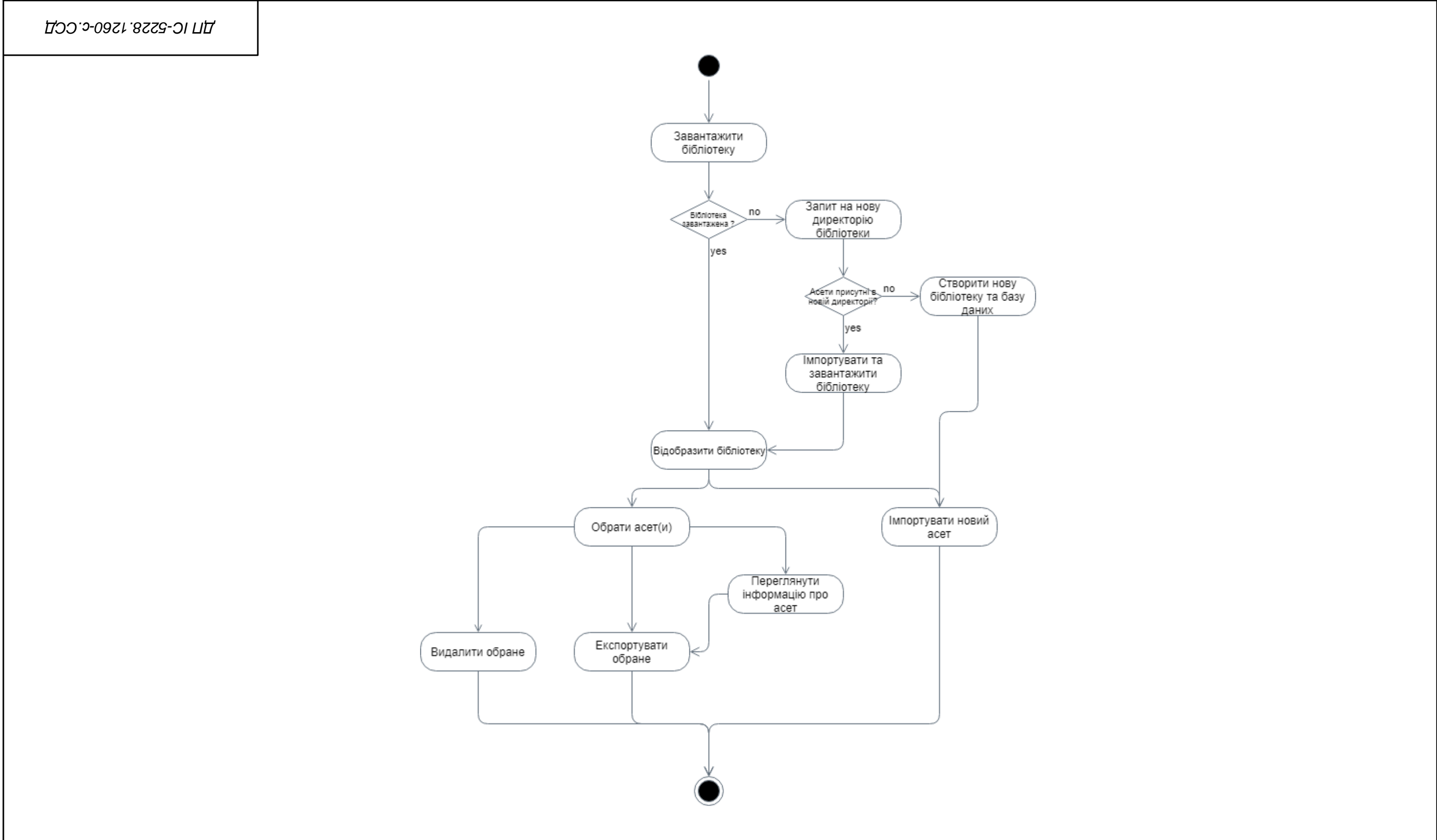


# **Графічний матеріал до дипломного проекту**

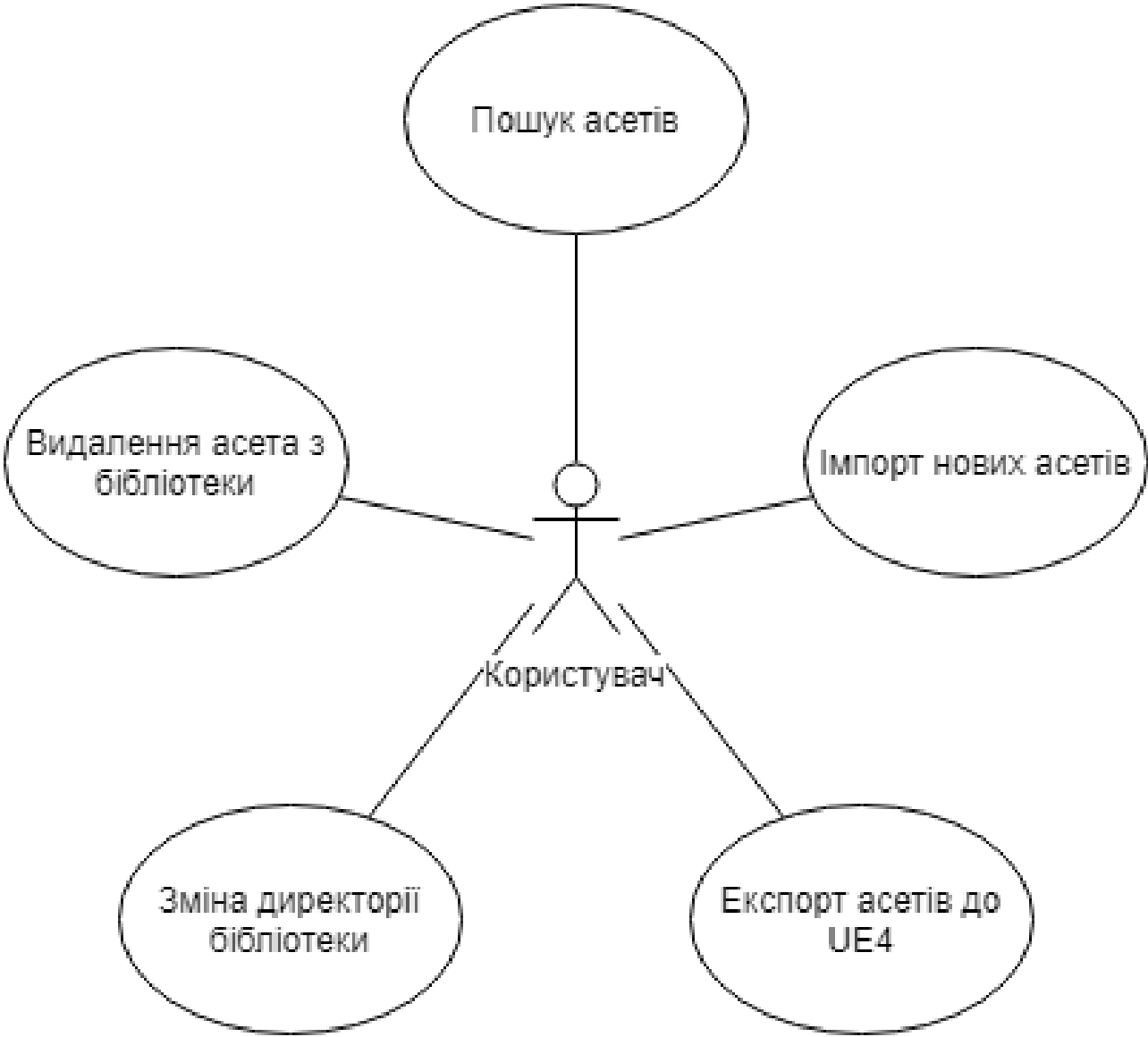
на тему: Інформаційна підтримка створення ієрархічної структури  
3D-об'єктів для 3D-сцен

---


Київ – 2019 року



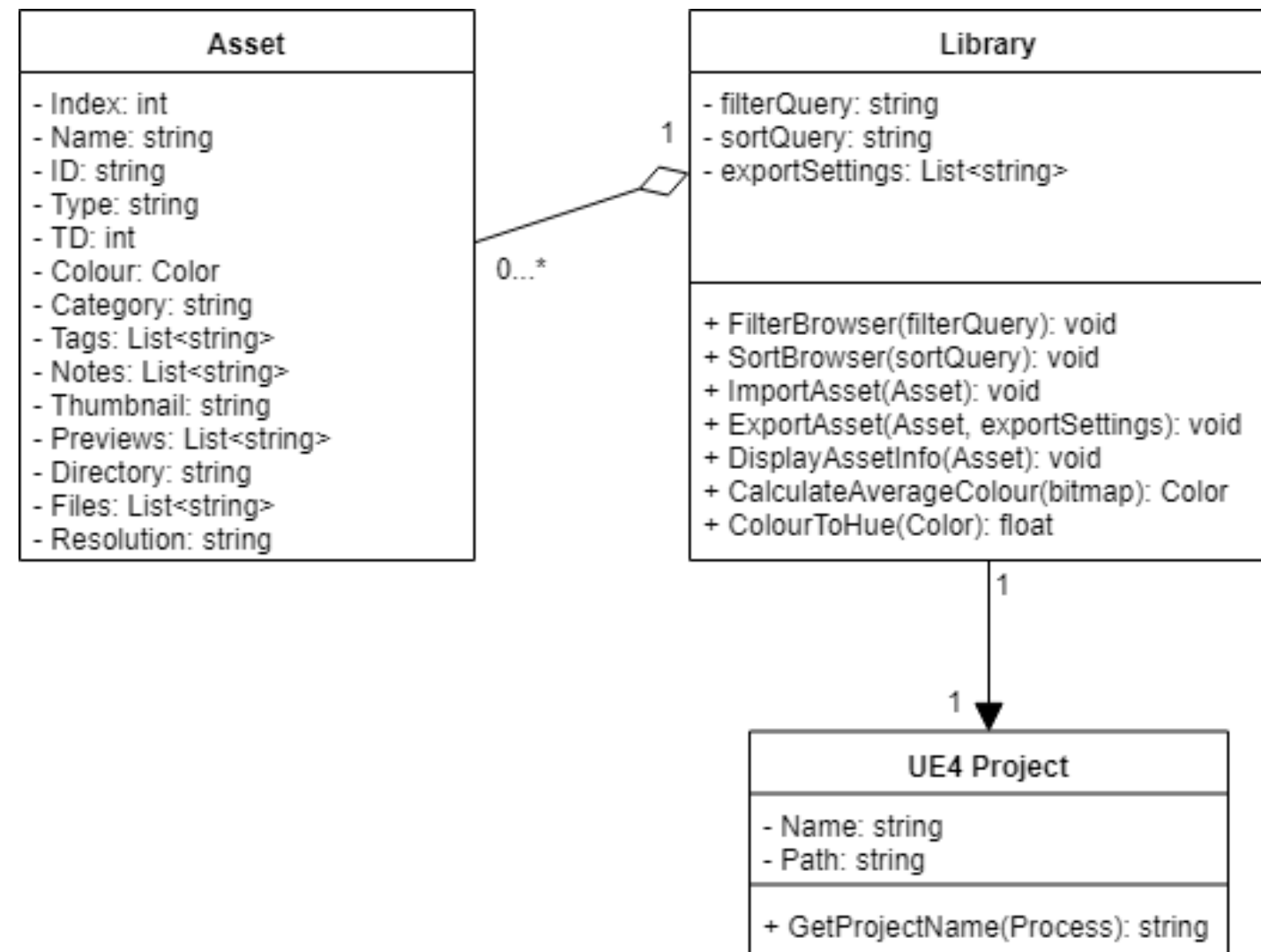
					ДП ІС-5228.1260-с.ССД				
					Схема структурна діяльності	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Юсеф А.							
Перевірив		ПроскураС.Л.							
Т. кон.						Аркуш 1		Аркушів 1	
					Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен		КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351		
Н. кон.		Телишева Т.О.							
Затвердив		ПроскураС.Л.							



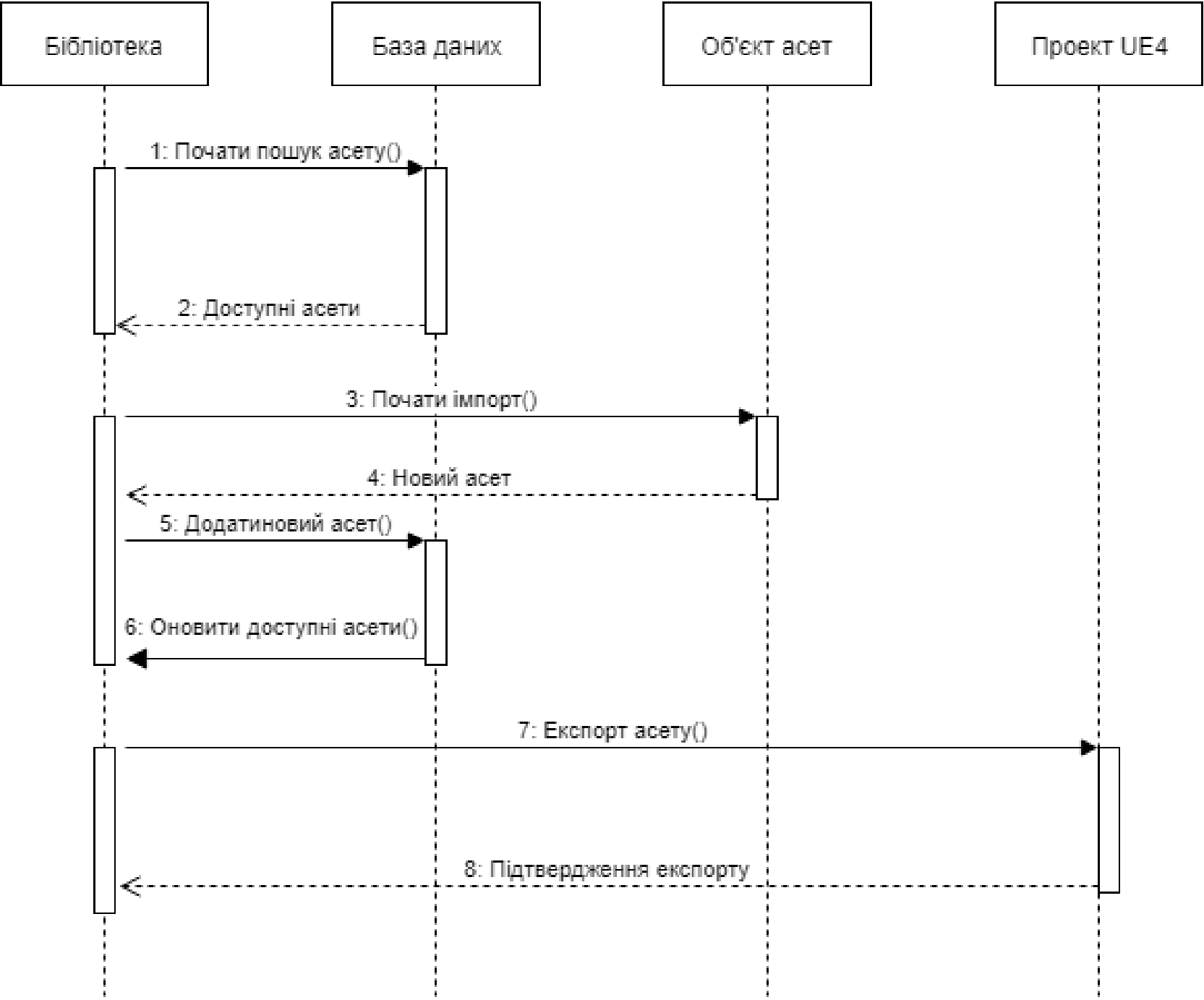
					ДП ІС-5228.1260-с.ССВ										
					Схема структурна варіантів використань					Літера		Маса	Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата											
Розробив	Юсеф А.														
Перевірив	ПроскураС.Л.														
Т. кон.										Аркуш 1			Аркушів 1		
					Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D- сцен					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351					
Н. кон.		Телишева Т.О.													
Затвердив		ПроскураС.Л.													

Assets	
id 	int
AssetName	string
AssetID	string
Type	string
TD	int
Colour	Colour
Category	string
Tags	string
Notes	string
Resolution	string
Hue	float

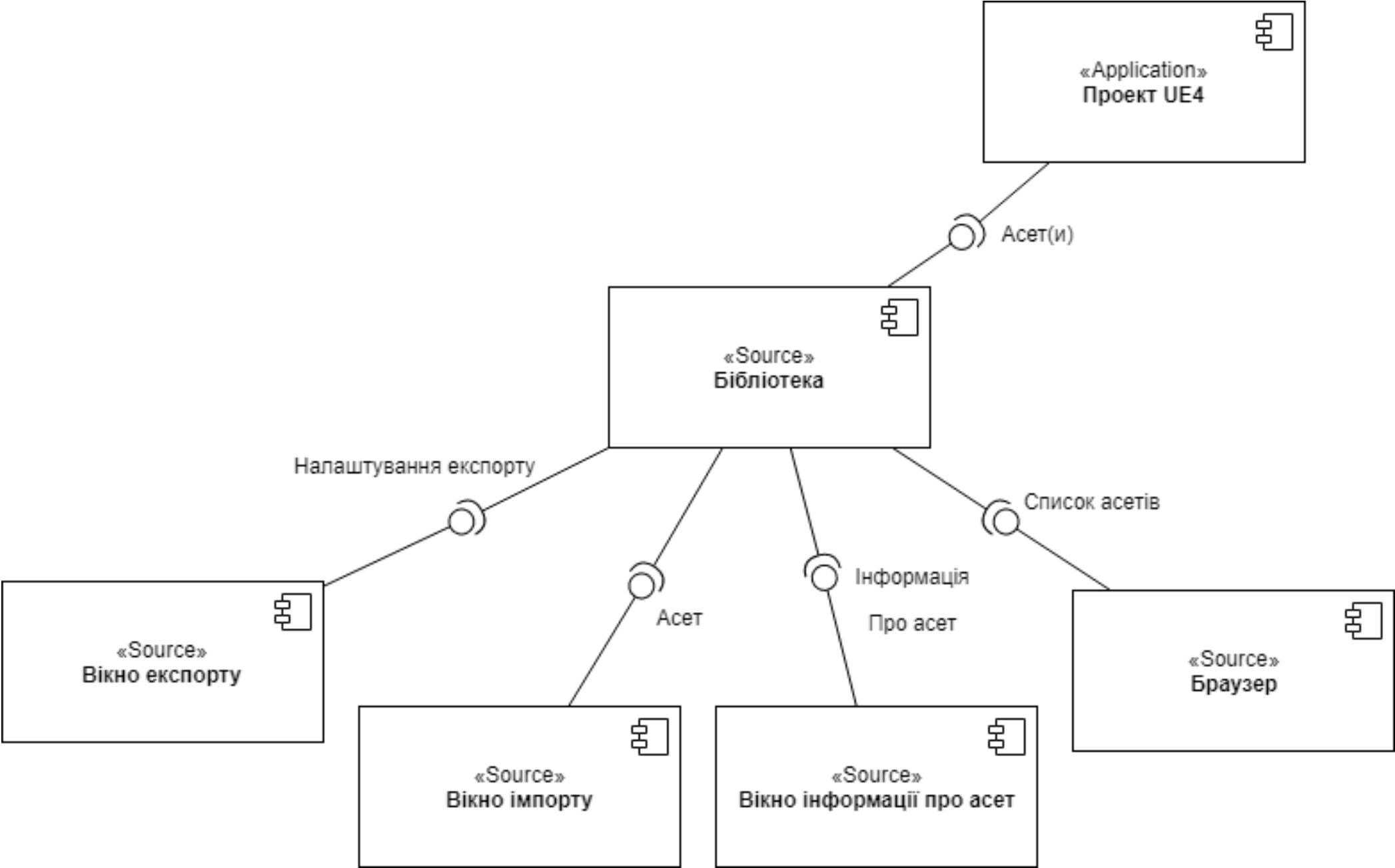
						ДП ІС-5228.1260-с.СБД					
	</										



					ДП ІС-5228.1260-с.ССК					
					Схема структурна класів програмного забезпечення	Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Юсеф А.									
Перевірив	ПроскураС.Л.									
Т. кон.						Аркуш 1		Аркушів 1		
					Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D- сцен	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351				
Н. кон.		Телишева Т.О.								
Затвердив		ПроскураС.Л.								



						ДП ІС-5228.1260-с.ССП						
						Схема структурна послідовності			Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Юсеф А.											
Перевірів	ПроскураС.Л.											
Т. кон.					Аркуш 1				Аркушів 1			
						Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D- сцен			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351			
Н. кон.	Тєлишева Т.О.											
Затвердив	ПроскураС.Л.											



					ДП ІС-5228.1260-с.ССК						
					Схема структурна компонентів програмного забезпечення	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Юсеф А.									
Перевірив		ПроскураС.Л.									
Т. кон.						Аркуш 1			Аркушів 1		
					Інформаційна підтримка створення ієрархічної структури 3D-об'єктів для 3D-сцен	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-351					
Н. кон.		Тєлишева Т.О.									
Затвердив		ПроскураС.Л.									